New Method of Electronic Control of Model Railroad Track

JAN HORÁČEK[1], JIŘÍ RYBIČKA[2], ROBERT ČÍŽEK[2]
[1]Faculty of Informatics,
Masaryk University,
Botanická 68a, 602 00 Brno,
CZECH REPUBLIC

[2]Department of Informatics,
Faculty of Business and Economics, Mendel University in Brno,
Zemědělská 1, 613 00 Brno,
CZECH REPUBLIC

Abstract: - The digital control of a model railroad track requires the control of the movement of vehicles and their functions (sounds, lighting) as well as control of trackside elements (switches, signals, occupancy detectors, etc.). While the control of the vehicles is provided by a standard whose functionality is perfectly adequate for normal operation, the control of the trackside elements can be carried out in several ways. For small domestic lines, there are commercial solutions from various companies, but for extensive (large) lines there is no such concept, and the appropriate solution is usually tailored to the application (club lines, simulators for training railroad staff). The tracks in the Railway Vehicles Control Laboratory of the Faculty of Business and Economics of Mendel University in Brno can be classified as large; therefore, a special system is required. However, the existing system is not suitable for several basic reasons. Therefore, a modernized system was developed. A new protocol for the RS485 bus was designed, the hardware and firmware of a new GPIO slave module MTB-UNI v4 was developed, a new RS485 master module MTB-USB was developed and two other new hardware modules including firmware for their MCUs were developed. A computer application MTB Daemon and a library for accessing the bus were developed. The result is a system that meets the relatively high requirements and enables further development and use in the laboratory.

Key-Words: - model railroad, electronic control, communication protocol, MTB, interlocking, Railway Vehicles Control Laboratory.

Received: June 29, 2022. Revised: September 26, 2023. Accepted: October 29, 2023. Available online: December 13, 2023.

1 Introduction

The Railway Vehicles Control Laboratory is a specialized workplace with two model railroads on an area of over 200 m^2. The model railroad is no longer a children's toy, but a large computer-controlled electronic complex. The main purpose of the laboratory is to be used in the teaching process, primarily as a teaching tool for program control of technological units and thus for programming in general. It is also used for diploma theses on various technical aspects of operation (automatic train detour, shuttle operation of motorized passenger trains, etc.). The track represents a digitally controlled technological unit whose design was adapted to the standards of the Model Railway Club Brno I (MRC Brno I), in whose cooperation the laboratory was founded 10 years ago and is operated today.

2 Current Status and Literature Review

The digital control of a relatively large layout requires different approaches than the control of a small "home" layout. Several sources mention model railroads that are used at universities or other institutions. Here, among other things, the control of trains and track elements is handled in different ways.

The article, [1], for example, presents the use of several alternative ways to control a model railroad. The system, whose functionality can be used, is based on PLC elements (Programmable Logic Controller), [2]. Although this article is somewhat older, it is still relevant to the concept of controlling model railroads. However, the fundamental problem with this solution for us is the purchase price. The basic element mentioned in this article, the AC 800M central control unit (type

PM860), is available at a price that is at least twice the annual budget for operating the entire laboratory. In a recent article, [3], where the details of the controller are presented, you will also find a photo showing the entire set of PLC elements.

Some interesting ideas can be found in the article, [4]. The system is focused on the operation of the train rather than the track elements, but the concept is based on the Arduino platform and supplemented with the necessary sensors. We have already discussed this approach in the design of a new control concept for a model railroad, where we saw the generality of the system as an advantage. On the other hand, the rather cumbersome structure of the required control electronics became apparent here, and it turned out that specialized elements are much more efficient.

One way to control the track elements is to connect them to the DCC control signal for the vehicles and control the track elements via the same bus. The article, [5], shows how to program a control station (OpenDCC) in a real-time operating system environment. We did not want to go down this route either, as bus throughput is a major limiting factor for large stations.

In the article, [6], the team of authors from the University of Applied Sciences Technikum Wien describes the structure of the so-called demonstrator that controls the model railroad. This is an approach in which the intelligence is distributed to the individual modules of the track. The system is controlled by MicroZed modules.

If we compare this approach with our MTB system, it is a completely different concept – here the intelligence is distributed, whereas in MTB it is concentrated in the computer.

The overview of possible approaches shows the solution options.

The basic criteria that a system for managing larger routes must fulfill are derived from, [7]. After some modifications for the needs of the laboratory under consideration, the criteria can be summarized in the following, partly interrelated points:

1. The ability to support hundreds of trackside elements (signals, switches, etc.) with multiple types of electrical interfaces.
2. Sustainability over time – the concept should be usable over several decades, i.e. based on the generic components that are expected to be available in their current or compatible form for a long time to come.
3. Extensibility of supported functions – new types of modules could be added in the future.
4. Acceptable financial requirements.
5. Two-way communication with the control computer, ability to recognize the correct functionality of the modules.
6. Independence from proprietary business solutions (for sustainability over time, but also acceptable financial requirements).
7. The ability to operate in a way that mimics a real railroad.

Although various electronic control components are available on the market, they are mostly intended for the general consumer, i.e. amateurs working in domestic conditions, and therefore usually do not meet some of the criteria defined above.

Let us briefly look at common solutions in practice.

2.1 Digital Command Control

Digital Command Control (DCC), [8], is probably the most widely used system for digital control of model railroads. There are no studies to confirm this, but it can be deduced from the number of components on the market. DCC is an open standard developed by the National Model Railroad Association (nmra.org). DCC standardizes the control of both trackside and mobile units.

Basic features: The main element of the system is the *command station*, which sends a DCC signal to control train movement and track accessories. The signal was designed as a unidirectional signal. A second bus, the so-called *feedback bus*, is used to receive the status of the elements on the track.

The bus is controlled by the command station, which queries feedback modules for data. The command station can be connected to a computer via another communication bus – the *throttle bus* (e.g. LocoNET, XpressNET). With the current state of the art, neither the vehicle nor the track-side decoders confirm receipt of the DCC command to the command station, which can lead to an error (non-delivery of the command or non-execution of the command for various reasons). The DCC trackside control scheme is shown in Figure 1.

There is no standard for how the feedback bus works, so different possibilities have been developed: S88, [10], RSbus, [11], or LocoNET, [12]. The latter is a licensed bus developed by Digitrax and its full use requires a license. However, it supports two-way communication and enables direct control of track devices without the need for a DCC signal.

A typical limitation of the DCC system is a relatively low maximum number of feedback modules. Extensive railroads usually solve this problem by dividing the line into sufficiently small,

isolated areas and operating independent modules in each area (e.g. applied to the very large line in the Railway Kingdom in Prague, presented at the 2015 technical excursion).

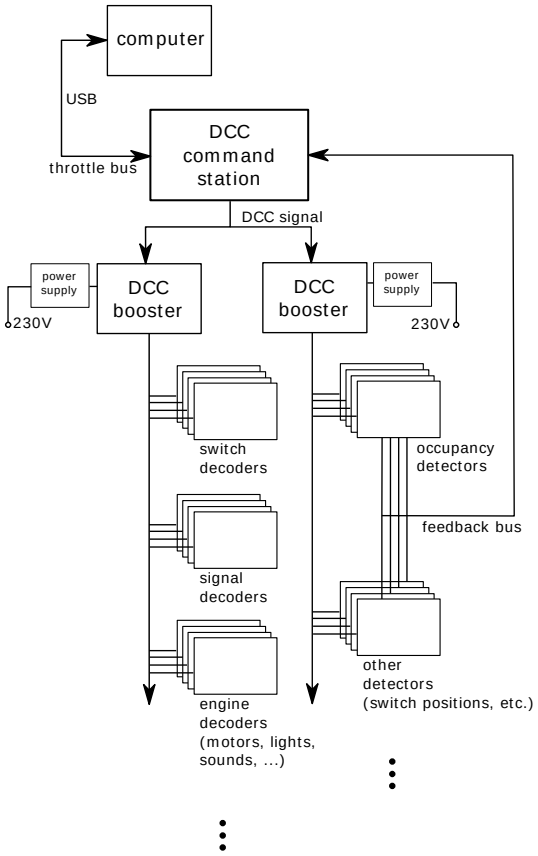

Fig. 1: Typical diagram of a DCC system, [9]

2.2 BiDiB

BiDiB (BiDirectional Bus) is an open bus developed by a community of model railroaders, [13]. It is based on RS485, there are different types of modules and the communication protocol is available online. However, BiDiB is only a protocol specification, there are no schematics, board designs, or firmware of BiDiB available. BiDiB fulfills all the criteria defined above.

2.3 MTB

The MTB v2 system (Model Train Bus), [14] is currently used in the laboratory. It enables the control of hundreds of trackside elements.

The control of the entire route is divided into two different areas:

- Vehicle control – is not addressed in the MTB system. For example, the DCC system with extended locomotive addresses can be used.
- Trackside (accessory) control – here, however, the MTB concept differs from the commercial concepts (the DCC system in commercial solutions usually also controls the accessories, as already mentioned in the corresponding section).

The separation of the accessory control from the vehicle control offers several advantages. Single communication bus *MTBbus* is used for both commanding track-side equipment and obtaining information from it, which allows for simpler (cheaper) design, own design brings high extensibility, fast bug fixing, independence on commercial manufacturers, and other benefits connected with in-house solutions.

Own design is not beneficial for vehicle control, because, in this area, open protocols exist, [8], thus risk of vendor-lock is minimized and the price of the components is quite low due to the competitive environment on the market.

It is then possible to build software support over both parts to allow to mimic the behavior of the signaling equipment on a real railway, which is a significant advantage for the teaching purposes of the laboratory.

2.4 Technical Description

A complex scheme incorporating both DCC and MTB systems is presented in Figure 2.

MTB system consists of:

1. MTBbus protocol specification. MTBbus is RS485 max. 115200 Bdps single-master bus.
2. MTB-USB module, which creates an interface between MTBbus and PC (via USB). It is master on MTBbus. It performs all real-time operations on MTBbus.
3. Specification of protocol between MTB-USB and PC.
4. MTB modules. There are various MTB v2 modules: MTB-UNI (universal – 16 digital inputs with IR point detectors support, 16 open-collector outputs), MTB-UNIm (same as MTB-UNI, but without IR support), MTB-TTL (16 digital inputs, 16 TTL outputs), MTB-REG (8 powerful analog outputs), MTB-POT (8 analog inputs).
5. Software support in PC (libraries for connecting to MTB-USB etc.).

More comprehensive specification of MTB v2 is available, [15], unfortunately in the Czech language only.

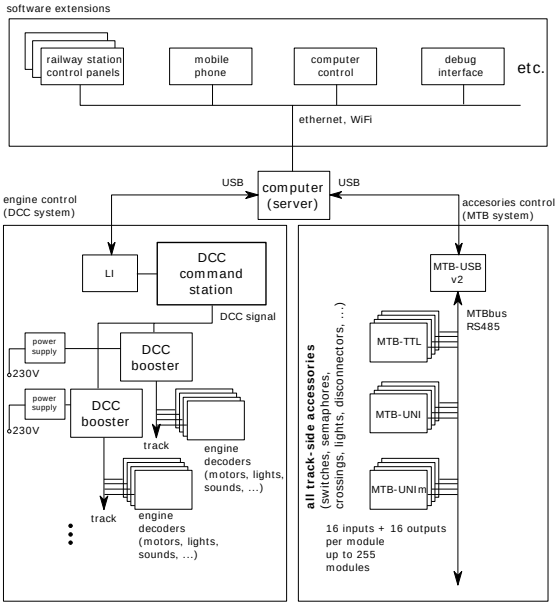

Fig. 2: Track control architecture in RVCL before modernization

2.4.1 Shortcomings of the Current MTB System

As complexly illustrated in Figure 2, the system is capable of controlling large tracks. The diagram describes the architecture of the railway in MRC Brno I and in the Railway Vehicles Control Laboratory of the FBE MENDELU. This allows the interchange of hardware, software, and know-how between the Club and the faculty.

Unfortunately, however, the accessory control part (MTB part) currently carries some very fundamental problems:

1. Outdated component base – the last update of the system was made in 2007 and some of the components used are currently unavailable. Installed electronics manufactured in the past and used on track have no direct replacement. This makes it impossible to expand the track with additional railways and stations, makes it impossible to replace broken parts of the electronics, and use electronics for other (e.g. diagnostic) purposes.
2. Unresolved licensing conditions – the authors of the system did not provide the source code for the firmware, electrical schematics, or PCB designs in a source form. Thus, essentially no updates can be made, bugs cannot be fixed, and new features cannot be added.
3. Technical capabilities of the processors – the current MTB system is based on *AT89C2051* MCUs with 2 kB of flash memory and 128 B SRAM. This capacity is insufficient for more complex applications. The firmware cannot be extended and the processors lack some key peripherals, e.g. EEPROM.

To enable the development of the laboratory it is necessary to modernize or replace the MTB system (the main purpose of the Internal Grant Agency project PEF_TP_2020004 on the Faculty of Business and Economic MENDELU). For this phase, the following general requirements have been defined resulting from the facts already mentioned:

1. The new system must be reasonably backward compatible with the current track control software.
2. The new system shall be sustainable for a sufficiently long time (estimated at 20 years).
3. The solution must not be costly.
4. Communication must include acknowledgment of controller actions; the correct functionality of the modules must be detected.
5. New functions and requirements defined in the future must be implementable, i.e. the system must be sufficiently extensible.

It is recommended to develop the new system as an open solution, i.e., hardware components as open hardware and all software as open source.

3 MTB v4

Based on the current situation described in the previous chapter, we decided to improve the MTB v2 system and design its new version *MTB v4*. We chose this solution because we perceive the advantages of the in-house solution as significant. Furthermore, own design allows students to understand the model railway control better, it allows them to experiment with it, develop alternatives, and simply transform their ideas into reality.

We chose not to use BiDiB because of the cost of deployment of the system. BiDiB modules in general are relatively cheap, however, the cost of replacing all the modules and wiring in the existing railway is unacceptable. MTB v2 must be simply upgradable to MTB v4.

Now we describe the design of the MTB v4 system. Full specification of MTB v4 is available at, [16].

3.1 Interfaces

On one side, MTB communicates with the track-side equipment (points, signals, sections, lights, etc.) – this must have been and was preserved. On the other side, MTB communicates with a computer. We keep using USB for this interface because we find no disadvantages to it. MTB v4

will be connected to interlocking software hJOP, [17], currently used in the laboratory. It connects to the track via a dynamically linked library with a defined API. New design must support multi-master control so that the individual elements can be controlled solely by the computer but with multiple applications. This design allows to connect e.g. both train interlocking hardware and street lights hardware to a single MTB module, which is then controlled by two different applications. This solution requires less hardware and wiring, which is highly desirable.

Until now, two types of outputs have been used for communication with the track-side hardware – digital binary (for all two-state outputs) and S-COM, [18], for transmission of signal aspects. In addition to these two, the new system adds the so-called oscillatory output (digital signal with a frequency in a small number of Hz) for indication purposes and special devices for train set uncoupling (uncouplers in the tracks). The problem of train uncoupling was dealt with in detail in the work, [19].

3.2 Modules

It was conceptually decided not to support analog inputs and outputs or pulse-width modulated outputs in MTB v4 because they are not needed in connected peripherals. In case they will be needed in the future, new modules will be developed.

MTB v2 uses three variants of universal IO modules – UNI, UNIm, and TTL – which differ in their support for infrared point sensors and in the type of outputs (TTL vs open-collector). To simplify and cheapen the solution, only one universal mass-producible module shall be developed. To connect specific peripherals, expansion modules can be used.

3.3 Other Improvements

In line with some of the above framework requirements, the new system should be able to detect modules that are added on the bus at runtime (not just at system start-up), as opposed to the current state. A problem in the same category is enabling easy detection of a malfunctioning module (e.g. during a power failure, disconnection of connectors, etc.). The innovation against the current state in this context appears to be the consistent implementation of command acknowledgment, which will enable the above functions.

For easy identification of a module in the track (individual modules are positioned on the underside of the track at the locations where the corresponding equipment is placed) it should be possible to switch on an indication LED on it from the control computer.

Updating module firmware should be possible directly over the MTBbus. Considering the number of modules on a large track (higher tenths or hundreds), it is very difficult to update the firmware manually. Using more advanced processors, this operation is achievable and allows updating of firmware on a module-by-module basis across the entire track. This entails the elimination of errors and omissions, including the elimination of time delays in manual updates.

3.4 MTBbus v4

The bus is designed from scratch. This is due to many significant changes compared to the current state, as well as the need to resolve licensing issues.

Hardware-wise, the RS485 standard is retained. The supported communication speeds remain three – 38 400 Bdps, 57 600 Bdps, and 115 200 Bdps. Lower speed is not necessary; at maximum speed, there are approximately 10 scans of each module per second with 100 modules on the bus (real packet length considered), which is enough for safety functions.

The communication is based on periodic polling of all modules on the bus, each module responds with a message. This confirms that the module is active. Active modules are polled more frequently than inactive modules to reduce latency. However, inactive modules must be included in the queries sometimes as well, so that they can be detected when new modules have been connected. Some ideas on how to create the communication protocol can be also found in the work of Lascano and Clyde, [20].

The message of MTBbus v4 consists of the following parts:

- Module address – 1 B, provides an address space for 255 modules, which is sufficient (address 0 is broadcast).
- Message length – as opposed to the MTB v2 situation where the maximum length could be 7 B, the length of the data part has been increased substantially, up to 120 B, which allows sending in addition to regular messages, firmware for module updates.
- Message code – a byte defining the meaning of the message.
- Message data part – up to 120 B of data.
- Checksum – compared to MTB v2, where only the XOR of the message is calculated, MTB v4 calculates a CRC-16 checksum instead. For longer message length in MTB v4, this type of

checksum is more appropriate. This solution is similar to the ModBus, [21], industrial bus.

The specification, [22], of MTBbus v4 is two-layered – the protocol defines messages, but the content of some messages is different for each MTB module type. The data format of inputs and outputs, the configuration format of the module, addressing, and memory organization for firmware updates are defined for each module type separately. This is advantageous because of possible further development, as the protocol can remain the same, although other types of modules with different characteristics may be added in the future without the need to modify the firmware of MTB-USB or the firmware of any older module.

Particular attention in the design is paid to firmware updates of MTB modules. The firmware update procedure is described in detail online in the document, [23]. The update can be done while the bus is fully operational, which is a great advantage especially if a module with a firmware of different original version is added or replaced.

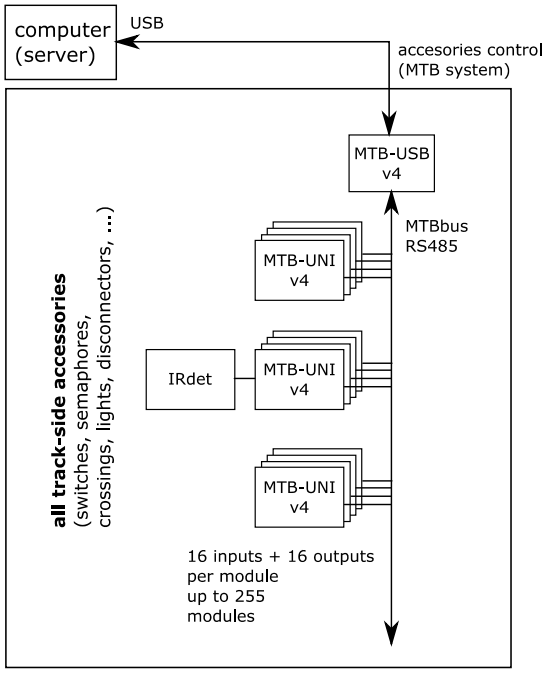

Fig. 3: A new concept of track control electronics – MTB v4 architecture

The flash memory in MTB modules can be overwritten from the bootloader. The update protocol reboots the processor into this program. Because it is unsafe to update the bootloader over the MTBbus, it must be a small and well-tested piece of firmware that will not need to be reprogrammed and will be loaded into the processor only once during manufacture by programming the module directly.

The changes in the design concept of the track control electronics are outlined in Figure 3. The implementation is described in, [7], in detail, here we briefly mention key parts.

3.5 MTB-UNI v4

Instead of the several types of MTB modules, only one type was designed with three freely adjustable options on each output pin: binary, S-COM, and oscillation mode. The *MTB-UNI v4,* [24], [25], universal module is built on modern components but with the expectation of long-term availability. MTB-UNI v4 is based on ATmega128A MCU, [26], and firmware is programmed in C.

The configuration of the module is permanently stored in the MCU because parameters like MTBbus speed or default output state must be applied immediately after power-on. However maintaining an authoritative configuration is handled by the computer, where all storage, backup, and versioning of configurations can be handled elegantly and much more easily. See section 3 for more details.

MTB-UNI v4 PCB design is based on well-known parts that are commonly used in industry. The emphasis is placed on its protection against damage from wrong wiring. The damage is done by connecting too high or too low voltage to the board. In that case, components may be burned. The protection is implemented with PTCs, thyristors, and Zener diodes. The design is based on a crowbar circuit, [27]. A big advantage in the design of the protection is its reversible cut-off – no fuses are burned.

MTB-UNI v4 firmware consists of a main firmware and a bootloader firmware. The bootloader is used for main firmware updating over MTBbus.

3.6 MTB-2-AVR

A cost-saving upgrade option was developed to keep the original MTB-UNI v2 modules, with the original processor being replaced with a new one (ATmega328P, [28]), and all other components, cabling, and mechanical parts remain. As there is not a new processor available that matches the pinout of the original processor, an add-on *MTB-2-AVR* board, [29], [30] was created to slide into the socket of the original processor. The add-on board carries the new processor in an SMD design with the necessary additional components. By simply replacing the processor in the original socket with the new board, a new MTB-UNI module is created.

MTB-2-AVR firmware consists of a main firmware and a bootloader firmware. The bootloader is used for main firmware updating over MTBbus.

3.7 IRdet

Infrared point sensor support had to be implemented to make the new solution deployable on the current and the new track where these sensors are used. An expansion board called *IRdet* was created to allow the connection of up to 8 IR sensors. Its outputs can be connected to the inputs of the MTB-UNI module. These outputs can also be used for other purposes (e.g. signaling in control panels).

3.8 MTB-USB v4

Implementation of the requirements forcing a different bus operation requires a new design of the *MTB-USB v4* module, [31], [32]. Here, backward compatibility with the original MTB-USB v2 board does not have to be strictly adhered to, as this board is only one throughout track and its complete replacement is neither technically nor financially too demanding.

PCB was designed in an open-source software KiCad, [33], design is based on MCU STM32F103, [34], and firmware is programmed in plain C. Firmware consists of a main firmware and a bootloader, which allows updating the main firmware via USB without the need for a special programmer (e.g. STlink, [35]).

4 Software

Two computer applications were developed as part of the solution – *MTB Daemon,* [36], and a library *hJOP MTB Network Library*, [37].

The requirement for multi-master control was solved in such a way that the MTB-USB v4 communicates with one application on the computer side – MTB Daemon –, but multiple control applications can connect to MTB Daemon via JSON REST API and command it. This functionality, among other things, allows students in the laboratory to program their simple applications controlling the tracks (IP-B project No. 8.1.17 of 2018). The programming language of MTB Daemon is C++17 with the Qt framework, which allows for a multi-platform solution; compilation is possible for both Linux-type OS, as well as Windows-type OS. Configuration of MTB Daemon and MTB modules is stored in human-readable form in JSON file. The file is under the version control system (git) separate for each deployment.

The hJOP MTB Network RCS Library interfaces the existing track control system hJOP and MTB Daemon. It is also developed in C++ as a DLL library.

5 Discussion

MTB v4 represents a solution for further development in the Railway Vehicles Control Laboratory. Expansions are planned to include additional stations, including the locomotive depot, which will allow to increase the operational possibilities and address other possible traffic situations, including various safety features and the implementation of the train timetable.

API of MTB Daemon allows students to create railroad-control applications without the need to struggle with hardware-related issues. Students can use programming languages independent of the implementation of the hardware and software components of the track. This contributes to an effective teaching of programming in the laboratory. In addition, the space has been expanded for students of specialized courses and thesis writers who can create applications based on MTB v4 and solve more challenging train control operation tasks.

API of MTB Daemon allows one to control a single bus with multiple control applications, which is unique on the market and useful, especially for students.

Compared to systems available commercially, the designed and implemented solution has a great advantage in the openness of the hardware and software, thus the parts needed by different users can be used without restrictions. In addition, the cost is lower than the components available for conventional domestic tracks. Here a new field of application is emerging – hardware components have been developed, but adaptation to commercial software systems has not yet been addressed. Support for MTBbus could be implemented in e.g. JMRI model railroad control software, [38], as a separate software module, or integration with e.g. TrainController, [39], commercial model railroad control software could be implemented via changing the PC interface of MTB-USB to the interface already supported by the TrainController software.

Compared to MTB v2 and other commercially available systems, MTB v4 brings a higher level of safety – all messages are acknowledged, module

failure is detected, etc. This creates pressure on the safety of similar commercial components.

In the future, the possibility of supporting higher bus speeds may be considered. This will allow an even shorter delay between the change of input of the MTB module and a reaction to the change in a computer. Another enhancement could be an automatic detection of the bus speed by MTB modules, important when connecting new modules or replacing broken modules on the fly. An interesting possibility could also be an extension to retransmit bus data over a wireless connection to control parts of the track that are physically separated from the control computer due to spatial, architectural, or other reasons. An example would be a railway consent console used in a foreign station at a modular tracks meeting, where each section of track is supplied by a different modeler but we need to operate the track as a whole.

Excluding already-mentioned latency and automatic bus speed detection, the last important limitation of MTB v4 is the maximum number of modules on one bus. From our point of view, 255 modules on one bus are enough, however, some non-standard applications of MTBbus that need more modules could be imagined. Simply adding more bits for module address is not suitable, as mechanisms like discovering new modules could take a long time and latency can grow considerably. A bus with a bigger maximum number of modules would probably require a different media access control mechanism.

6 Conclusion

MTB v4 has been implemented and deployed on all tracks in the Railway Vehicles Control Laboratory as well as on all tracks in MRC Brno I. That includes 3 independent layouts with 103 MTB modules in total. Most of these are MTB-UNI v2 modules with MTB-2-AVR add-on modules. 7 MTB-UNI v4 modules have been in operation for 100+ hours each without any major problems. MTBbus v4 has been in operation for 300+ hours in total with only minor difficulties. These issues were solved with new firmware versions, where uploading a new firmware over MTBbus turned out to be very useful.

MTB v4 can also be used in other fields, where centralized computer control of several dislocated peripherals is used. Examples include home automatization, production line controlling, or remote monitoring. Generally, MTB is a cheaper variant suitable for applications, where PLCs are currently used.

Acknowledgments:
A major contribution to modernization was made by the thesis of Jan Horáček.

References:

[1] Pavlas, R. Digital Control of the Laboratory Railways Model with a PLC Automat. In Farana, Smutný, Kočí & Babiuch (eds.) XXXII. Seminar ASR '2007 "*Instruments and Control*", pp.193–196. VŠB-TUO, Ostrava, 2007. ISBN: 978-80-248-1272-4.

[2] ABB. *PLC automation*, [Online]. https://new.abb.com/plc (Accessed Date: November 10, 2023).

[3] Pavlas, R., Zavadil, J. Control system of small model railway. Four ways of tracking control. *13th International Carpathian Control Conference (ICCC)*, IEEE, 2012, pp. 533–536. ISBN: 978-1-4577-1868-7.

[4] Rahman, S., Kawser, A., Rumman, K. M., Rahman, F., Rubab, A. Design and Implementation of Intelligent Railway System. *Journal of Image Processing and Intelligent Remote Sensing*, Vol. 02, No. 05, pp. 53–62, Aug-Sept 2022, https://doi.org/10.55529/jipirs.25.53.62.

[5] Pok-Son Kim, Kutzner, A. Digital Model Railway Control on the Foundation of Real-Time Operating systems. *Information*, vol. 17, No. 1, pp. 241–250, 2014. ISSN: 1343-4500.

[6] Rössler, P., Höller, R., Schrön, F., Reisner, C., Ewers, E. A Model Railway based Demonstrator for Safety-Critical Systems. In *Proceedings of the 12th European Workshop on Microelectronics Education (EWME)*, 2018, pp. 24–28. ISBN: 978-1-5386-1157-9.

[7] Horáček, J. *Design and implementation of a new MTBbus protocol* [*Návrh a implementace nového protokolu sběrnice MTBbus*] Diploma thesis. Supervisor: Zdeněk Matěj. Masaryk University, Brno, 2021.

[8] NMRA. *Electrical Standards for Digital Command Control*, [Online]. https://www.nmra.org/sites/default/files/standards/sandrp/pdf/s-9.1_electrical_standards_for_digital_command_control_2021.pdf Last rev. Apr 9, 2021 (Accessed Date: December 17, 2021).

[9] Wikipedia. *Digital Command Control*, [Online]. https://en.wikipedia.org/wiki/Digital_Comma

nd_Control (Accessed Date: December 17, 2021).

[10] OPENDCC. *Transmission of S88 data over network cable*, [Online]. https://www.opendcc.de/s88/s88_n/s88-n_e.html (Accessed Date: April 26, 2021).

[11] Pras, A. *RSbus Arduino Library*, [Online]. https://github.com/aikopras/RSbus. (Accessed Date: April 26, 2021).

[12] DCC WIKI. *LocoNet overview*, [Online]. https://dccwiki.com/LocoNet (Accessed Date: April 26, 2021).

[13] BiDiB. *Short summary of BiDiB [Kurzzusammendfassung von BiDiB]*, [Online]. http://bidib.org/protokoll/intro.html (Accessed Date: April 26, 2021).

[14] Báňa, V., Trávník, P. *MTB modules communication protocol [Komunikační protokol modulů MTB]*, [Online]. https://mtb.kmz-brno.cz/assets/pdf/mtb-protok20.pdf (Accessed Date: April 28, 2021).

[15] Horáček, J. *MTB v2* [online] Available at https://mtb.kmz-brno.cz/cz/v2 (Accessed Date: October 10, 2023).

[16] Horáček, J. *MTB – IO bus for controlling accessories of model railways*, [Online]. https://mtb.kmz-brno.cz/ (Accessed Date: October 10, 2023).

[17] Horáček, J. *hJOP: RCS system*, [Online]. https://hjop.kmz-brno.cz/rcs (Accessed Date: May 3, 2021).

[18] Báňa, V. *S-Com signal decoder [Dekodér S-com pro světelná návěstidla ČSD]*, [Online]. https://www.mtb-model.com/elektro/s-com-nav.htm (Accessed Date: April 20, 2021).

[19] Galieva, Z. *Automation bypassing of trains on railway model*. [*Automatické objíždění souprav na modelovém kolejišti.*] Diploma thesis. Supervisor: Jiří Rybička. Mendel University in Brno. 2020.

[20] Lascano, J. E., Clyde, S. W. A Pattern Language for Application-level Communication Protocols, [Online]. In: *The Eleventh International Conference on Software Engineering Advances.* 2016. ISBN 978-1-61208-498-5. https://www.academia.edu/31103209/A_Pattern_Language_for_Application_level_Communication_Protocols (Accessed Date: April 20, 2021).

[21] Modbus Organization Inc. 2021. *ModBus Application Protocol Specification*, [Online]. https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf (Accessed Date: May 4, 2021).

[22] Horáček, J. *MTBbus v4.1 Protocol Specification*, [Online]. https://github.com/kmzbrnoI/mtbbus-protocol (Accessed Date: October 10, 2023).

[23] Horáček, J. *MTBbus workflows*, [Online]. https://github.com/kmzbrnoI/mtbbus-protocol/blob/master/workflows.md (Accessed Date: October 10, 2023).

[24] Horáček, J., Čížek, R. *MTB-UNI v4 PCB*, [Online]. https://github.com/kmzbrnoI/mtb-uni-4-ele (Accessed Date: October 10, 2023).

[25] Horáček, J. *MTB-UNI v4 Firmware*, [Online]. https://github.com/kmzbrnoI/mtb-uni-4-fw (Accessed Date: October 10, 2023).

[26] Microchip Technology. *ATmega128A*, [Online]. https://www.microchip.com/en-us/product/atmega128a (Accessed Date: April 20, 2021).

[27] Electronic notes. *SCR Thyristor Crowbar: overvoltage protection circuit*, [Online]. https://www.electronics-notes.com/articles/analogue_circuits/thyristor-scr-triac/overvoltage-protection-crowbar-circuit.php (Accessed Date: April 20, 2021).

[28] Microchip Technology. *ATmega328P*, [Online].https://www.microchip.com/en-us/product/atmega328p (Accessed Date: September 20, 2021).

[29] Horáček, J. *MTB-2-AVR PCB*, [Online]. https://github.com/kmzbrnoI/mtb-2-avr-pcb (Accessed Date: October 10, 2023).

[30] Horáček, J. *MTB-2-AVR Firmware*, [Online]. https://github.com/kmzbrnoI/mtb-2-avr-fw (Accessed Date: October 10, 2023).

[31] Horáček, J. *MTB-USB v4 PCB*, [Online]. https://github.com/kmzbrnoI/mtb-usb-4-pcb, (Accessed Date: October 10, 2023).

[32] Horáček, J. *MTB-USB v4 Firmware*, [Online]. https://github.com/kmzbrnoI/mtb-usb-4-fw (Accessed Date: October 10, 2023).

[33] Kicad. *KiCad ESA – A Cross Platform and Open Source Electronics Design Automation Suite,* 2023, [Online]. https://www.kicad.org/ (Accessed Date: April 20, 2021).

[34] ST Microelectronics. *STM32F103* [online]. Available at https://www.st.com/en/microcontrollers-microprocessors/stm32f103.html (Accessed Date: September 20, 2023).

[35] ST Microelectronics. *ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32*, [Online]. https://www.st.com/en/development-tools/st-

link-v2.html (Accessed Date: September 20, 2023).
[36] Horáček, J. *MTB daemon*, [Online]. https://github.com/kmzbrnoI/mtb-daemon (Accessed Date: October 10, 2023).
[37] Horáček, J. *hJOP MTB Network RCS Library*, [Online]. https://github.com/kmzbrnoI/mtb-net-lib (Accessed Date: October 10, 2023).
[38] JMRI Community. *What is JMRI?* , [Online]. https://www.jmri.org/ (Accessed Date: August 13, 2023).
[39] Freiwald Software. *Model Railroad Computer Control with TrainController™*, [Online]. https://www.freiwald.com/pages/traincontroller.htm (Accessed Date: August 20, 2023).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Jan Horáček made a main contribution to the concept, design, and implementation of the final solution, implementing software, PCBs, and firmware.
- Jiří Rybička participated in the concept and final testing.
- Robert Čížek participated in circuit design.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The upgrade of the MTB system in the Railway Vehicles Control Laboratory was funded by the IGA team project No. PEF_TP_2020004 called “Innovation Electronic Control of Model Tracks”.

Conflict of Interest

The authors have no conflicts of interest to declare.