# NEW DIDO BUS USED IN A SPECIAL AND HARD INDUSTRY CONDITION

**LEOS BERAN, TOMAS MYSLIVEC**

Faculty of Mechatronics, Informatics and Interdisciplinary Studies

Technical university of Liberec, Liberec, Czech Republic

**DOI: 10.17973/MMSJ.2021_03_2020077**

**leos.beran@tul.cz**

This paper presents a new serial asynchronous Bus, using standardized digital input and output from a PLC system for the hardware layer. The data frame and time setting are described to illustrate the need to establish a new serial bus for study. Our selected implementation within a warehouse robot is presented as our industrial case study. It is our study case in the industry. This presented bus can be used in many different situations with similar needs or requirements. This Bus is not designed for sending large data with high throughput.

**KEYWORDS**

serial bus, asynchronous, PLC, digital input/output, hard conditions, automation, carbon brush.

## 1 INTRODUCTION

Many different industrial serial buses have been developed in the past. Each of them has a different purpose and application. Some were designed for point to point (RS232, EnDat, Io-Link, I2C) [Wilkening 2020], [Debayan 2010], [Mussolini 2019], communication while others were for multipoint communication within a net (RS485, CAN) [Qiangsheng 2010], [Wang 2013], [Simonik 2014].

Contemporary trends in the 21st century tend to faster data communication between many different users. The most recent standardization within the automation industry is the OPC UA standard, which can connect different machines in one net. This is intended mainly for Industrie 4.0 [Li 2020].

People need to have everything faster, bigger and better. This is the natural outcome of research and development from the very beginning of mankind. If systems are faster, they are more efficient and more profitable. This has an influence on our everyday lives.

However, there are some situations where slower can be better for certain technical solutions. The price of HW (Receiver and Transceiver) is also an important factor in many technical applications. Many of the above-mentioned busses need both very good metallic and shielded connections between the transceiver (TRS) and receiver (REC) for flawless communication. There are many HW configurations that cannot fulfil this requirement in the industry. In these applications, it is not possible to use standard industrial serial busses.

It is not easy to present slower bus communication in the 21st century. However, we would like to do it. Typical HW configurations where standard buses have problems operating without errors can be defined:

a) a connection between TRS and REC is through unshielded wires

b) communication pairs (wires) are placed in a parallel, not twisted, configuration

c) there is a sliding contact (carbon brush-ring) in the connection

d) it is necessary to handle more commands via two wires

Common buses are usually very sophisticated and complicated in their implementation. They do have their own place in the industry of course where they are very useful and efficient. It is not our goal to defame other buses. However, we would like to present our solution for a digital input/output bus which is both very simple and robust.

## 2 HARDWARE BUS DEFINITION

A new DIDO bus, based on our design, is an asynchronous digital duplex serial communication bus. The name DIDO comes from digital input (DI) and output (DO). Input is used as a receiver and output is used as a transceiver. Figure 1 shows a possible connection diagram for the two PLCs.

The physical layer from the ISO/OSI reference model is based on three wires (TxD, RxD and GND). The connection can be established using different types of the conductive path with different cross-sections and shapes of conductors. The conductors can be wires, planchettes, rods, carbon brushes, rings etc.

The electrical definitions **log 1** and **log 0** are taken from PLC logical levels. Any voltage lower than 5 V is log 0 (FALSE) and any voltage higher than 15 V to 24 V is log 1 (TRUE). A standard input or output from any PLC can be used thanks to this definition. All applications in the automation industry usually have some DI and DO. This means there are no other fees for the used HW.

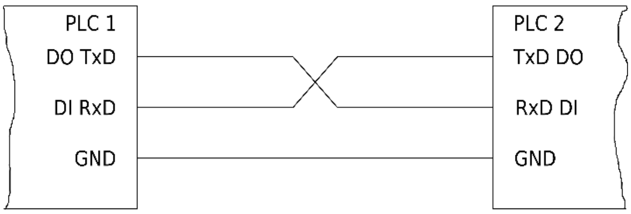


**Figure 1.** Connection diagram PLC1 and PLC2

The time cycle ***Tc*** can be set for the actual HW configuration. It depends on the concrete DI a DO switching frequency. In our case, it is 10 ms for one bit. The idle state of the bus is log 1. If log 0 is longer than *Tc**10 ms, the transceiver or connecting wire is defective.

The maximum length of the connection depends on the resistance of the conductor that connects the TRS and REC. Maximum drop voltage can not be more than 9 V. In this case, the ones sent via the bus would not be detected as 1 (TRUE). A transceiver's power supply must be sufficient to prevent a voltage drop while transmitting data, which could lead to a false interpretation of that data.

| | Udn [V] | Uup [V] |
|---|---|---|
| **log 1** | 15 | 24 |
| **log 0** | 0 | 5 |

**Table 1**. *Voltage level for log 0 and 1*

There are some similarities with the RS232 standard, which was used mainly as the null modem between two PCs in the past. The RS232 can be found in many industrial applications these days.

It still has its own place. If we were to use it in our applications, we would have to have an RS232 device in our PLC. Furthermore, RS232 needs for its proper operation a good cable connection to operate properly. The lowest baud rate is usually mentioned 2,4 kbaud. It means that one bit takes 0,4 ms. This time is too short for our intended purpose.

## 3 DATA FRAME DEFINITION

The link layer, from the ISO/OSI reference model, is a data frame that is sent over the DIDO in three main parts. These parts can be seen in Figure 2.

The first starting part has a three bits sequence 0-1-0. The first edge 1->0 is a synchronization bit for a receiver. The second and the third bits 1-0, show that the change on the line is not caused by electromagnetic disturbance.

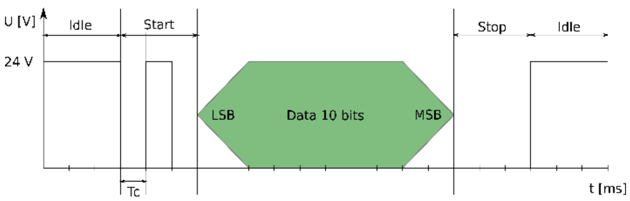


**Figure 2**. Time diagram of data frame

*The second* part is intended for data that should be sent from a source to a destination. The size of the data part is 10 bits. Each of them can be determined for TRUE/FALSE information. This means you can send 10 information statements ON/OFF or you can send 10bits value in the range from 0-1023. It is possible to define the data part as needed. If an application needs more data, the frame can be extended or the data to be sent can be divided into fragments of 10 bits.

*The third* ending part has a three bits sequence 0-0-0. After this 3**Tc* level voltage at 0 V is communication finished. Both TRS and REC then wait for the next sequence to start.

There are no CRC bits in the data frame. Our implementation verified the physical layer is robustly designed to prevent any errors in data that is sent via this bus. They can be added as optional in between Data a Stop part if needed.

## 4 TRANSCEIVERS IMPLEMENTATION FOR A PLC

Hardware implementation is very easy when using DO from a PLC. DO must be connected with a DI (RxD).

The software implementation can be carried out using different programming languages. We used Structured Text for its clarity. The code is put into two basic states WAIT_CMD and SEND_FRAME. This is shown in Figure 3. In the first state, the data frame is assembled according to the chosen command/data. It is 16 bits variable in our case. The commands can be stored in predefined constants for better clarity.

If the data is ready, then the state is changed to SEND_FRAME where the prepared data from BufferToSend are sent one bit by one to the serial line. The time base is set to *Tc*. Our *Tc* is 10 ms. Our data frame is sent in 160 ms. This is a really long time compared to other buses. However, it is the main reason the DIDO bus is so reliable within a difficult industrial environment. We would like to have a faster time for sending the data but it would be necessary to consider switching the frequency of the DO and the minimum task class time of the used PLC.

This timing constraint must be considered when you create a design of your technical solution. It is not possible to have the information from a sensor faster than 160 ms for example. It is the greatest disadvantage of this solution.

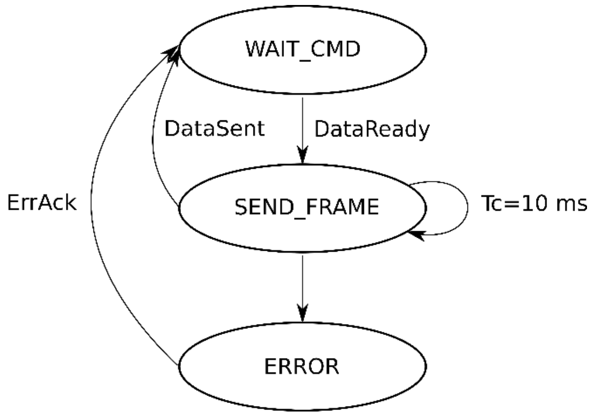


**Figure 3**. Transceiver - state machine diagram

## 5 RECEIVER IMPLEMENTATION FOR A PLC

Implementing the receiver is a little bit complicated compared to the transceiver. The algorithm must make samples during the listening period and, after that, must find the data that was sent. The sampling period must be at least Tc/5. This means we can get 5 samples for each bit that was sent.

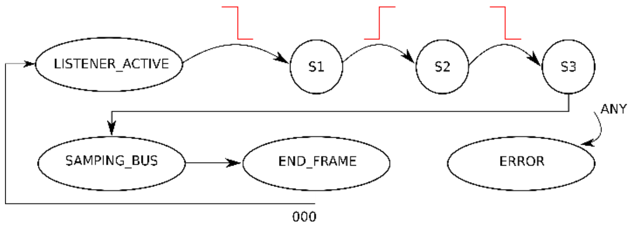


**Figure 4**. Receiver - state machine diagram

The state machine in Figure 4 has seven states for receiving the complete frame. In the state LISTENER_ACTIVE the algorithm is waiting for a starting edge. This starts checking the states sequence 0-1-0 in states S1, S2 and S3. If the starting sequence is not correct the state machine sets the ERROR state. If the starting sequence is correct, the sampling of received data continues into the SAMPLING_BUS state (the green dots in Figure 5). During the sampling period, data is decoded and put into the `RecievedData` variable.

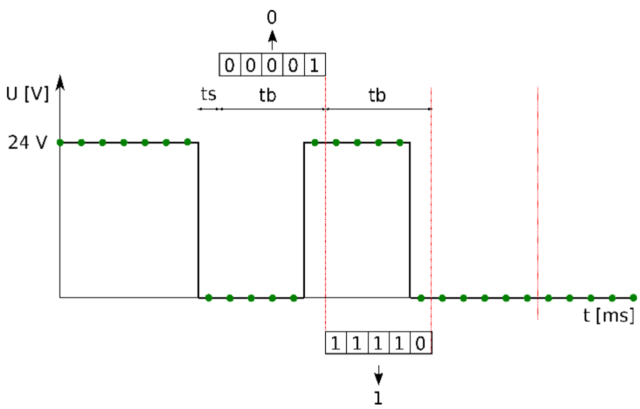


**Figure 5**. Sampling time diagram of a received signal

The acquired samples are processed in groups of five. In this group are evaluated zeros (FALSE) and ones (TRUE). If there are more ones than zeros samples, the bit is saved as TRUE, and vice versa. This evaluation was chosen because of a possible time shift **ts** between TRS and REC. The maximum ts is 2*ts. Otherwise, the evaluation of received data would be defective.

It is not necessary to have any buffer for received data. The system (PLC) is much faster than the DIDO bus in processing received data immediately. If necessary, a buffer can be added for later processing.

## 6 IMPLEMENTATION THE DIDO BUS IN ATMEGA328P

Both receiver and transceiver can be implemented on different platforms. We implemented the second end of the DIDO bus using an ATmega328P microcontroller with an Arduino bootloader and professional two-phase stepper motor driver TB6600. Thanks to this implementation, we are able to control the stepper motor in the intended application.

The software implementation in Atmega328 is very similar to PLC, but we used a different programming language (C++).

The main program is designed as a loop supplemented by three interrupts. Two of these are from end switches and signal the RxD, the last one is from the `Timer1`.
Considering that the microcontroller ATmega328P has only two external interrupts, there is a simple way to use only one pin interrupt for the two sensors.
The easy circuit with diode uses connection signals from both end sensors to a single pin, see Figure 6. The sensor logic needs software modification.

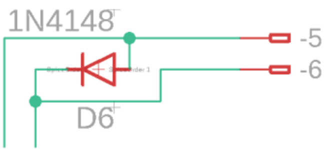


**Figure 6**. Schema of both end sensors to the single pin

The design of the pin RxD is solved as a simple stabilizer with a Zener diode. It reduces and stabilizes voltage from 24 V to 5 V (TTL logic). The circuit diagram can be seen in Figure 7.

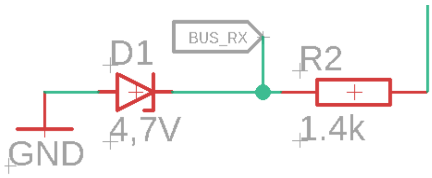


**Figure 7**. Schema of stabilizer with the Zener diode

The pin TxD of the board uses an optocoupler for switching 24 V (HTL logic). This converts TTL to HTL logic. Figure 8. shows the connection optocoupler on the board.

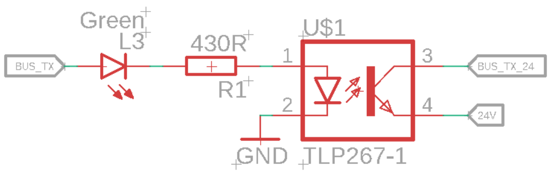


**Figure 8**. Schema of TTL to HTL converter with optocoupler

We did not find any problems with this board during our system testing. The circuit with the ATmega328P and the TB6600 stepper motor driver appears to be a reliable solution for this application.

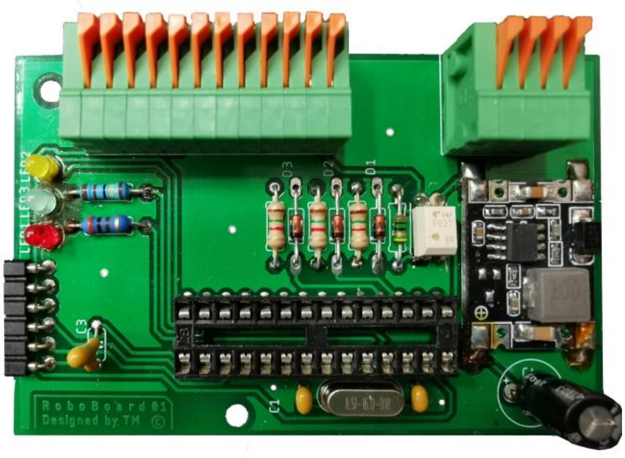

**Figure 9**. Prototype control unit for microcontroller ATmega328P

The application is very simple. It can carry out OPEN (MOVE FWD) and CLOSE (MOVE BWD) commands. The stepper motor can move between two limit switches, which determine the working range of the mechanism.

## 7 EXPERIMENTAL WORKPLACE

Our schematic diagram of the experimental workplace can be seen in Figure 10. below. It consists of two main parts, with one (7-main body), is stable while the other (4-elevator) can move up and down.

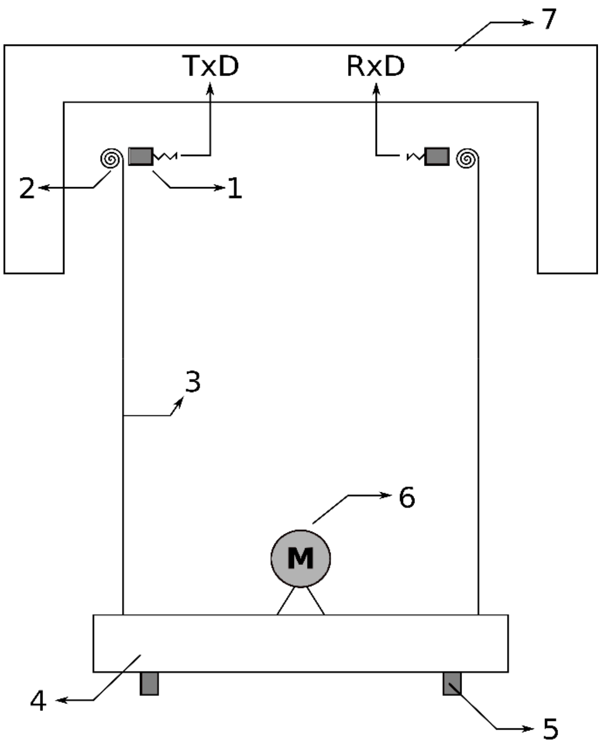


**Figure 10**. Experimental workplace (1-carbon brush, 2-winder, 3-planchette, 4-elevator, 5-sensor, 6-stepper motor, 7-main body)

These two parts are connected with four planchettes (4) made from stainless steel. Two of these connect the TxD and RxD signals. The other pair connect the 24 V DC power supply. The maximum distance between the parts is 8 meters. The floor plan in Figure 11. shows the fastening places in the second part (4).
This configuration of wires (planchettes) is very inconvenient from the point of view of electromagnetic compatibility. It can create a large “antenna” for disturbing electromagnetic waves.

However, very similar situations and configurations can also be found in the industry.

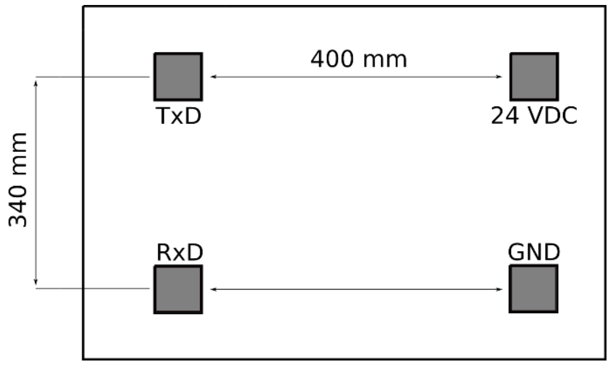


**Figure 11**. Fastening configuration of planchettes

Additionally, there are four carbon brush (1) and ring (2) connections on each line. These are not appropriate for signal transmission and can cause a disturbance. All in all, it is not good for both the mechanical and electrical configuration of standard serial buses. This was the main reason why we started to develop our own communication bus.
The motor (6) represents an actuator for manipulating the mechanism. It is the stepper motor in our setup.

The practical application of our DIDO bus was for a special warehouse robot that could manipulate crates (the brown one in the illustration), with goods stored under its platform. The robot is seen in Figure 13. The mechanical configuration corresponds to Figure 10. This robot can pick up any crate from a depth of up to 8m. The moving velocity of a crate is up to 1,3 m/s. The concept of the warehouse is pictured below in Figure 12. There are two ports for operators and three robots which can manipulate any crate stored within the determined space of the warehouse.

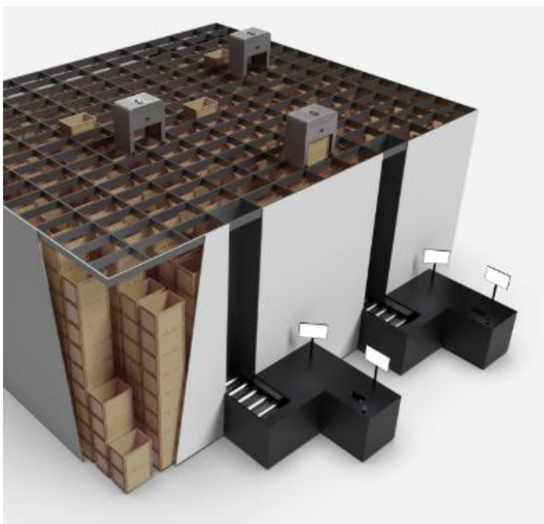

**Figure 12**. Warehouse concept

After picking up a crate, the robot can move it to any point within the warehouse prior to processing within the factory. The lifting of the crate must be very reliable. The stepper motor on the board handles the gripper, which fastens the crate to the board, which then lifts it.
Two inductive sensors work as positive and negative limit switches creating the positional limits of the mechanism.

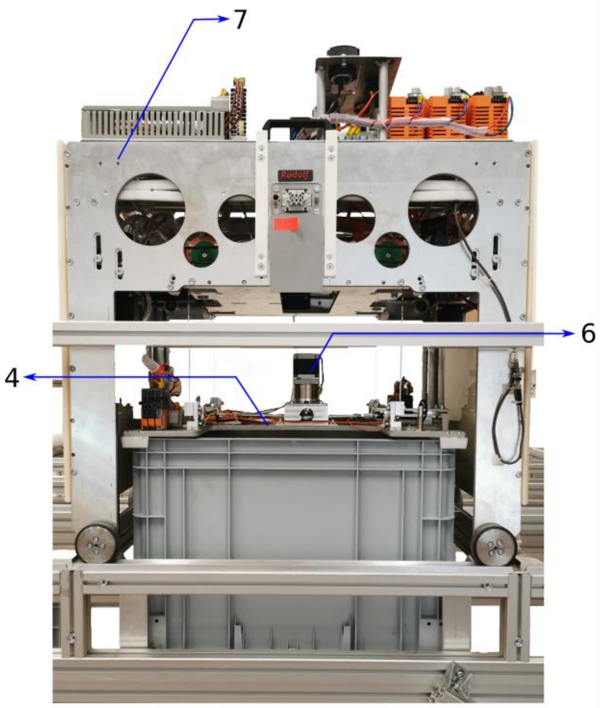


**Figure 13**. Real application of DIDO bus

Both command and state information are communicated, through planchettes and carbon brush contacts, as quickly as possible. Our test featured DIDO implementation in the first version of this application. The robot was able to carry out 25 thousand picks up/lay down cycles within six days with just three errors in communication. This can be considered a satisfactory result for the first implemented version. The next version will cover these errors.

## 8 NEXT POSSIBLE INTENDED APPLICATIONS OF DIDO BUS

Other applications can be found where there is a need to operate actuators or sensors for rotating parts. These parts are usually connected using a carbon brush and ring. This type of connection can cause many signal disturbances.

### *8.1 Printing head application*

Additive technologies are becoming more common in many different branches of the industry. The construction building industry is very important for people, but it is also very conservative. We can see clear developments in this area over the last five years. There are now many different projects in the world that want to build a house using a 3D printer.
The printing head used to print cement mixtures usually needs rotating nozzles for this purpose. If we need to measure or adjust some part(s) of these nozzles, we need some connection with their sensors or actuators. Now we have our own bus for connecting these necessary parts for the intended technology.

### *8.2 Wheel sensing application*

A very similar situation can be seen in the automotive industry. There is a need to measure values connected with wheels, for example, the air pressure in a tire, the deformation of a rim, the temperature from a brake system etc. Manufacturers usually use wireless communication for these purposes. If you need to use an actuator that needs more power, you have to use a carbon brush and ring connection. Our solution can be used here without any problems.

## 9 CONCLUSION

Both software and hardware DIDO bus implementations were presented in this paper. The results show that the new design can be used in industry with very hard conditions considering the conductive way. Furthermore the wires TxD and RxD can be in a parallel configuration that creates a huge wire loop. This loop has a bad effect on a signal in a disturbing magnetic field.

Some possible applications were presented to imagine some another for a better understanding of our goal. Our DIDO bus can be used for many different purposes that we do not see now. The future will show other concrete applications.

It can be claimed that the proposed solution is very robust and useful for real industrial applications. Our solution was verified in laboratory conditions. This first version will be tested in a pilot plant soon. New results will bring a knowledge base to improve this version. The next improved version will be ready to use for upcoming projects in the future.

It can be claimed at the end that our solution creates the possibility of transferring a small amount of data via a bad connection caused by carbon brush - ring contact and parallel wires (planchettes) with a large distance between these wires.

**ACKNOWLEDGMENTS**

The research presented in this paper/article was supported by the European Structural and Investment Funds of the Operational Programme Research, Development and Education within the project "3D Print in civil engineering and architecture - 3D STAR" reg. number CZ.02.1.01/0.0/0.0/16_025/0007424.

The result was obtained through the financial support of the Ministry of Education, Youth and Sports of the Czech Republic and the European Union (European Structural and Investment Funds - Operational Programme Research, Development and Education) in the frames of the project “Modular platform for autonomous chassis of specialized electric vehicles for freight and equipment transportation”, Reg. No. CZ.02.1.01/0.0/0.0/16_025/0007293.

**REFERENCES**

[Debayan 2010], Debayan, P. at al. Implementation Of An IO-Link Interface Library Component For SoC Applications. In: IET Irish Signals and Systems Conference 2010 (ISSC 2010), 2010. Cork: Institution of Engineering and Technology, pp 222-227. ISBN:9781617820571.

[Mussolini 2019] Mussolini T. at al. Implementation of an I2C to Profibus Serial Comunication Interface. In: *2019 MIXDES - 26th International Conference "Mixed Design of Integrated Circuits and Systems"*. 2019. Rzeszów: pp. 390-393.

[Simonik 2014], Simonik P. at al. Principles and techniques for analysis of automotive communication lines and buses. In: 2014 ELEKTRO, 2014. Rajecke Teplice: pp. 500-503.

[Li 2020], Li Y. at al. Practical Implementation of an OPC UA TSN Communication Architecture for a Manufacturing System. In *IEEE Access*, vol. 8, pp. 200100-200111.

[Qiangsheng 2010] Qiangsheng, Y. Research and Application of CAN and LIN Bus in Automobile Network System. In: 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 2010. Chendgdu: pp V6-150 - V6-154.

[Wang 2013] Wang, Li. at al. Design of STM32-Based CANopen Motion Control Master in Transfer Robot. In: 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, 2013. Shenyang: pp 1609-1612.

[Wilkening 2020] Wilkening T. at al. EnDat 3 – Safety-Related Fully Digital Encoder Interface from the Application Point of View. In: PCIM Europe digital days 2020; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, 2020. Germany: pp. 1-8.

**CONTACTS**:

MSc. Leos Beran, PhD.
Technical University of Liberec, Faculty of Mechatronics, Informatics and Interdisciplinary Studies
Studentská 1402/2, Liberec 1, 461 17, Czech Republic
+420 485 353 772, leos.beran@tul.cz, www.tul.cz