IO-Link Master Bridge

David Matoušek
Department of Electrical Engineering
University of Pardubice
Pardubice, Czech Republic
david.matousek@upce.cz

Ondřej Černý

Wikov MGI
Hronov, Czech Republic
ocerny@wikov.com

2022 International Conference on Applied Electronics (AE) | 978-1-6654-9482-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/AE54730.2022.9920068

***Abstract*—This article describes the practical implementation of the IO-Link master bridge, which ensure an interface between IO-Link devices and a main controlling computer. The main effect of this implementation is a simple data acquisition system based on a microcontroller that enables an easy expansion of many connected IO-Link devices. Concerning universal usage, the communication between the IO-Link master bridge and the computer works on standard UART units. Main computer and IO-Link master bridges interconnection ensure optocouplers. Thus, this method guarantees galvanic isolation firstly and easy expansion by the open collector technique secondly.**

Keywords*—*IO-Link, IO-Link master, UART, microcontroller.

I. Introduction

IO-Link is a serial, bi-directional point-to-point connection for signal transmission and energy supply under any networks, fieldbuses, or backplane buses. An IO-Link system consists of the following basic components: IO-Link master, IO-Link device (e.g., sensors, valves, motor starters, I/O modules), unshielded 3- or 5-conductor standard cables, and engineering tool for configuring and assigning parameters of IO-Link [1].

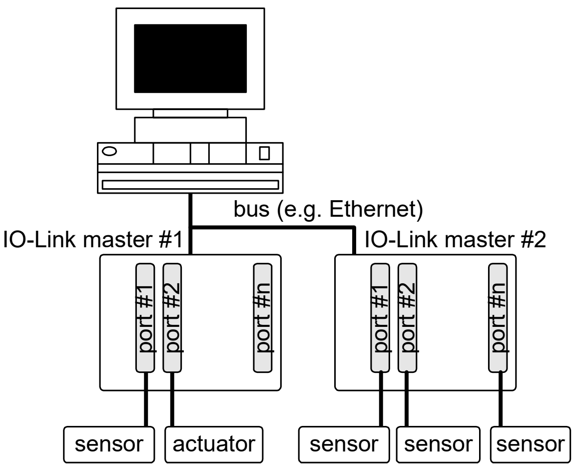

Fig. 1. Simplified example of IO-Link system architecture.

The main advantages of the IO-Link system are:

- Standardized uniform interface for sensors and actuators irrespective of their complexity.
- Consistent communication between sensors/actuators and the controller. Access to all process data, diagnostic data, and device information. Remote diagnostics supported.
- Integrated device identification.

The IO-Link system uses a 4-pin plug connector for sensors and a 5-pin plug connector for actuators. Meanwhile, The IO-Link master uses a 5-pin socket connector. The main pins are 24V (power), 0V, and C/Q (switching and communication line) [2].

The IO-Link transmission works as a half-duplex asynchronous serial channel. The transmission rate is standardized as COM1 (4.8kb/s), COM2 (38.4kb/s), and COM3 (230.4kb/s).

The IO-Link system uses four data types: process data (e.g. data measured by the sensor), value status (indicates the validity of process data), device data (parameters, identification data, and diagnostic information), and events (signalization of error).

II. Genesis of IO-Link Master Bridge

The need to create your own IO-Link master arose in solving the project “New generation of online monitoring for the gearbox diagnostic with the help of artificial intelligence”. In solving this project, it turned out to be advantageous to replace a sizable number of different sensors (temperature, pressure, wear, etc.) working on varied communication buses (current loop, RS-485, etc.) with IO-Link type sensors, which usually sense more quantities.

At the time of the project, there was no simple and inexpensive converter between the IO-Link device and the computer. A relatively detailed state of the art of available solution took place before the design of its solution. Available development kits are MAXREFDES165# ($ 600, four IO-Link ports), STEVAL-VP004V2 ($ 156, four IO-Link ports, limited use of 10000 minutes), P-NUCLEO-IOM01M1 ($ 68, only one IO-Link port, limited use of 10000 minutes). None of the above solutions allows easy scalability (link multiple IO-Link master units to some serial communication bus or port).

Therefore, we have accepted the need to design an interface for connecting IO-Link devices to a computer using a commonly available communication channel. Thus, we determined the following parameters of the IO-Link master bridge:

- The computer and the bridge interconnection warrants galvanic isolation.
- The bridge communicates via an asynchronous serial channel (saving of connecting wires, easy communication control – the bridge can send a response with a delay controlled by a time-out).
- The bridge will connect at least four IO-Link devices with the possibility of connecting more bridges in parallel (thus increasing the number of connectable IO-Link devices).

This work has been supported by the Technology Agency of the Czech Republic within the TREND Programme, project no. FW03010244.

- Concerning price and availability (due to the chip crisis), we will choose a microcontroller with a sufficient number of serial channels (each IO-Link device needs one serial channel). Alternatively, the microcontroller will scan sensors using the time-multiplex method.
- In the current version, we will only support IO-Link sensor devices operating at a fixed baud rate of 38.4kb/s.

Concerning availability, we chose the microcontroller ATmega328PB [3] and IO-Link driver LTC2874 [4]. Primarily, we considered STM32 or PIC32 series microcontrollers (they have a higher number of serial channels), but their availability was very poor. In the end, the choice of the 8-bit ATmega328PB microcontroller proved to be a great idea. This microcontroller has only two serial channels. We used channel #0 to communicate with the master computer and channel #1 to communicate with the LTC2874 driver. The LTC2874 integrated circuit allows you to create an IO-Link master unit with four channels. It is just an intelligent voltage level converter supplemented by current limiting circuits. Thus, the microcontroller solved all communication activities. LTC2874 has won the performance-price ratio compared to similar products [5], [6].

We solved LTC2874 channel switching using a 74HCT151 [7] multiplexer. This solution has only one problem. Namely, the microcontroller cannot read data from the connected IO-Link devices simultaneously. However, this was not a limitation for our purposes. The readings of the data from the sensors took place once a one to five seconds.

The system uses a 24V DC power supply. IO-Link devices need this power supply directly. The logic uses 5V DC power supply that creates DC/DC converter. The control computer uses a separate 3.3V DC power supply. We solved the galvanic separation by using optocouplers for RxD and TxD lines.

The final circuit solution is depicted in Fig. 2.

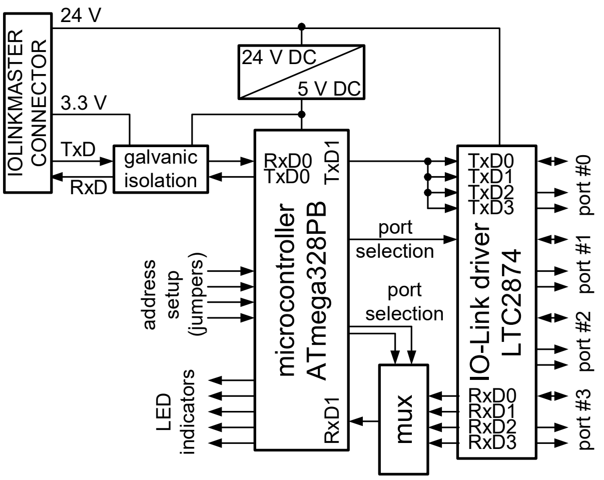

Fig. 2. Block diagram of the four-channel IO-Link master bridge.

In the design phase of the IO-Link master bridge, we addressed the issue of the future interconnection of several bridges. As the simplest solution, we chose the open-collector technique (known from the I^{2}C type serial buses). The open-collector technique allows connecting optocoupler outputs for data transmitted from microcontrollers to the control computer. Meanwhile, the interconnection of lines for receiving data by microcontrollers is not a problem.

The described way of passing the control computer and microcontrollers is evident from Fig. 3. At the top are the RxD (data reception) and TxD (data transmission) lines of the control computer (there is also a separate 3.3V DC supply). On the right is the connection of two IO-Link master bridges.

We solved the mutual distinction of individual bridges by addressing. Each IO-Link master bridge has its unique address in the range 1 to 15 (address 0 is reserved). The address is defined by four jumpers (see Fig. 2). The handshake between the control computer and the bridges ensures that the control computer communicates with only one bridge in a moment, which it selects by sending the corresponding address.

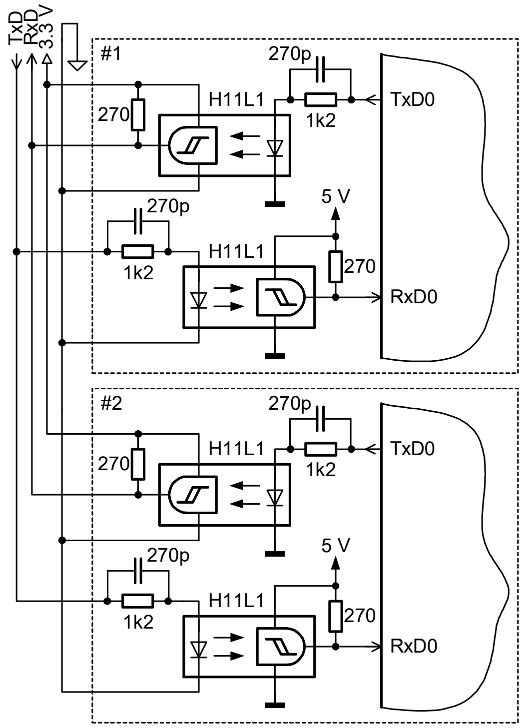

Fig. 3. Schematic diagram of interconnection between the computer and two IO-Link master bridges.

III. Communication Protocol

The communication between the control computer and the IO-Link master bridge proceeds via a serial channel (speed 38.4kbps, 8 data bits, no parity, one stop-bit). The checksum ensures data integrity. Communication protocol defines five commands for data exchange with IO-Link sensors:

- #0: find out the bridge type and firmware version.
- #1: find out the used ports.
- #2: read sensor parameters on the specified port.
- #3: read sensor process data on the specified port.
- #4: reset the sensor event on the specified port.

The communication starts with the control computer, which sends two bytes. These bytes include the IO-Link master bridge address, the port number of the sensor used, the command number and the checksum (see Fig. 4). IO-Link master bridge sends a response whose format depends on the command used. However, the first byte informs about the number of bytes that follow, and the last byte is the control sum.

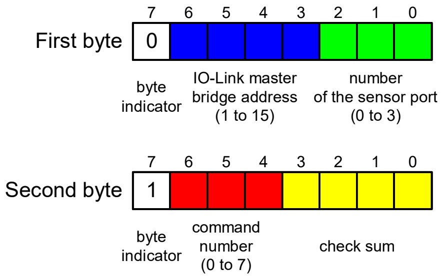

Fig. 4. Format of a command of the control computer.

A. Command #0

Use this command to specify the board type and firmware version.

The example below shows sending a bridge command #0 with address 5 (see Fig. 5). The bridge responds that the bridge version is 1, and the firmware version is 1. The first byte of the response says that two bytes follow. The last byte of the bridge response is a checksum.

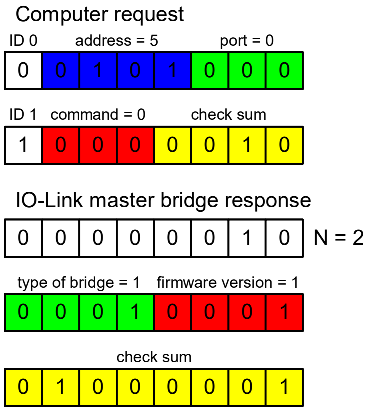

Fig. 5. Format of the command #0.

B. Command #1

Use this command to specify which bridge ports are engaged by IO-Link sensors.

The computer sends for example binary values 0b00101000 (address = 5, port = 0) and 0b10010011 (command = 1, check sum). The bridge responds in binary form 0b00000010 (N = 2), 0b00000111 (IO-Link sensors connected on ports #2, #1, #0) and 0b01010111 (check sum).

C. Command #2

You can use this command to retrieve sensor parameters. This command determines the type of the sensor. Parameters allow correct interpretation of the process data from this sensor.

The computer sends for example binary values 0b00101010 (address = 5, port = 2) and 0b10100010 (command = 2, check sum). The bridge responds (described in symbolic form) N (14 bytes), MinCycleTime, FrameCapability, IOLinkRevisionID, ProcessDataIn, ProcessDataOut, VendorID1, VendorID2, DeviceID1, DeviceID2, DeviceID3, FunctionID1, FunctionID2, unpacked process data length, and check sum. The VendorID1, VendorID2, DeviceID1, DeviceID2, DeviceID3 parameters allow uniquely determine the type of the sensor. This information will then allow us to interpret the process data of this sensor.

D. Command #3

Control computer sends command #3 to obtain sensor process data and event information. The interpretation of the received data depends on the type of sensor (determined by command #2).

The example below shows sending a bridge command #3 with address 5 (see Fig. 6). The bridge sends process data and event code. This example assumes that the sensor uses two-byte process data.

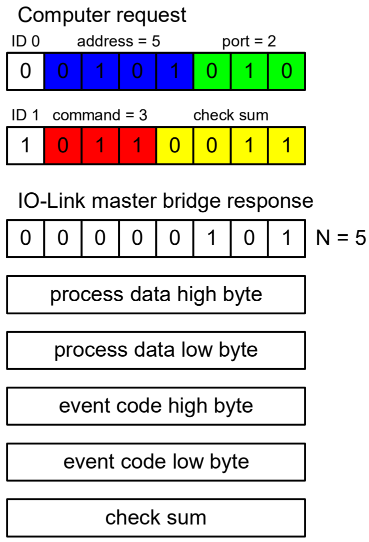

Fig. 6. Format of the command #3.

E. Command #4

If the sensor has detected an event (for example, a "deviation" of the measured value from the specified limits), we obtain a non-zero event code when using command #3. The event code comes until it is manually reset. Command #4 resets the event code.

The computer sends for example binary values 0b00101010 (address = 5, port = 2) and 0b11000100 (command = 4, check sum). The bridge resets the event on the specified sensor but doesn't respond.

IV. Firmware

The LTC2874 IC performs signal voltage matching only. Thus, the microcontroller controls all communication processes. When the program starts, the microcontroller initializes all used peripherals firstly. Then the LED displays the selected communication address. Finally, the program enters an infinite loop in which it reads data from individual sensors.

The timing of the sensor communication time-outs solves a free-running timer #1 with a resolution of 500 ms. The timer #3 controls the indicator LED. Each LED indicates the status of one sensor. The LED lights continuously when the sensor communicates successfully. If the sensor is disconnected or a communication error has occurred, the corresponding LED flashes. The USART #0 performs communication between the microcontroller and the control computer. The USART #1 performs communication between the microcontroller and the

IO-Link sensor. The SPI #1 is used to set up the LTC2874 circuit.

The main program loop saves the flag of success communication for each sensor. This flag is evaluated as the result of previous communication. If the communication was successful, the microcontroller reads sensors parameters. This action is followed by a reading of the process data. The program stores the parameters and processes data of each sensor in a separate block of memory. If the communication fails, the program will try to initialize the sensor by a maximum of three repetitions of the wake-up command [8]. To avoid unnecessary CPU blocking when performing this operation, time-outs are measured using the global variable named time via appropriate interrupt vector. This variable is controlled by timer #1 via the appropriate interrupt vector.

Communication between the control computer and the microcontroller is performed by USART #0. This communication is asynchronous to the main program loop and uses receive and transmit interrupt vectors. Thus, the main program loop mines the sensor data independently of the communication with the control computer. The acquired parameters and process data of the sensors rest in memory until the control computer picks them up. If the control computer does not communicate, the sensor data is updated automatically. Thus, actual sensor data is always available in the memory.

CONCLUSION

As part of the TAČR TREND project - FW03010244 “New generation of online monitoring for the gearbox diagnostic with the help of artificial intelligence”, an IO-Link master bridge was designed, which allows the connection of up to four IO-Link sensors (See Fig. 7).

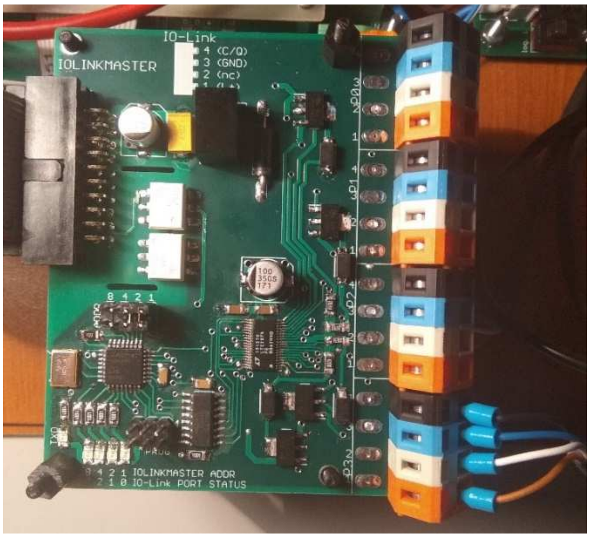

Fig. 7. IO-Link master bridge 4-layer PCB (top view).

The system integrator can easily connect IO-Link master bridge boards in parallel. Thus, more than four sensors can be connected (see Fig. 8). In total, up to 15 boards (with addresses 1 to 15) can be connected in this way (address 0 is reserved for other purposes). We could not be verified this assumption practically because such amount sensors were not available during testing. At least we verified this assumption in another connection using the same optocouplers. The number of boards is limited only by the current capability of the TxD signal (the driver must supply more than 20 mA for full load).

Fig. 8. Three IO-link master bridges with unique addresses in the sandwich.

The fact that each board contains a separate control and communication processor means that the data acquisition takes place for each independently of the others. Sensors connected on the same board are addressed sequentially, but sensors connected between different boards are addressed simultaneously.

The scan period of the board is 500 ms. We chose this interval concerning the expected reading speed of the sensor process data once per second. We read the process data of the sensors using interrupt. Thus, there is virtually no time lag between the request to read the sensor data (the data is always current) and the corresponding response.

The used communication speed of 38.4 kbps theoretically ensures the throughput of command #3 (obtain sensor data) around 480 samples/s per sensor (if we consider 2-byte sensor data). For example, we can read data from four sensors (whether sensors are on one board or each sensor is connected to a separate boards) 120 times per second. We can say that the selected communication speed is high enough for a higher number of sensors. If we read the process data of the sensors faster than the scanning period allows, we get the same data repeatedly. However, this is normal behavior of IO-Link master units.

The IO-Link master bridge is a significantly cheaper solution (about $ 60) than using commercially available IO-Link masters. It is also advantageous for our application that it communicates via a classic serial port, which is commonly equipped with control computers and microcontrollers.

REFERENCES

[1] IO-Link Community, “IO-Link System Description,” www.io-link.com, 2018.

[2] IO-Link Community, “IO-Link Interface and System,” www.io-link.com, June 2019.

[3] Microchip Technology Inc., “ATmega328PB Datasheet Complete DS40001906C,” www.mcrochip.com, 2018.

[4] Linear Technology Corporation, “LTC2874 Datasheet LT 0515 REV B,” www.analog.com, 2013.

[5] J. DeAngelis and M. Jackson, “IO-LINK HANDBOOK, 2nd ed.” www.maximintegrated.com, 2020.

[6] STMicroelectronics, “Products and solutions for IO-Link technology,” www.st.com, 2020.

[7] Texas Instruments Incorporated, “CD54HC151, CD74HC151, CD54HCT151, CD74HCT151 Datasheet SCHS150D,” www.ti.com, November 2021.

[8] J. Uffelmann, P. Wienzek, and Dr. M. Jahn, IO-Link The DNA of Industry 4.0, 1st ed.. Essen: Vulkan-Verlag, 2019.