embedded world 2013
Exhibition & Conference
... it´s a smarter world

IO-Link Protocol Stack for a Configurable Microprocessor

Johannes Röttig, Missing Link Electronics
Endric Schubert, Missing Link Electronics
Markus Tahedl, ifm electronic gmbh

Paper ID: 434

Session: M2M Communication II

Date: 28 February 2013

Abstract:

Sensors can be turned into smart sensors by implementing powerful signal processing and control algorithms. Adding a bidirectional data link unleashes the full possibilities of smart sensors. IO-Link is the new industry standard for such connectivity. It enables sensor configuration over the product lifecycle and adds powerful failure detection and diagnostic methods. In combination with microcontroller architectures embedded inside modern Field-Programmable Gate-Arrays, this adds additional layers of system configurability.

Introduction

In the global machine market customers expect reliable machines that quickly deliver a return-on-investment [1]. Process monitoring and control is important to ensure flexibility, production process optimization and highly available operation. With the integration of data links it becomes possible to get direct access to sensors and actuators. Additional information and control options are available, today. Standard field bus systems like Profibus or AS-Interface are well known. However, the implementation of field bus modules may be only viable for expensive sensors. Therefore the simple and cost-efficient communication standard IO-Link was defined. It is field bus independent and can be integrated into all fieldbus systems.

Direct access to sensor and actuator data as well as universal data structures for extended process control and monitoring enables the configuration of terminal equipment in the machine. With configurable devices the necessary flexibility is provided under changing environmental conditions. Even changing requirements for different applications are manageable in the equipment lifecycle with dynamic configuration strategies.

With the availability of failure detection methods and diagnostic information selective maintenance is possible. For example, wiring errors due to disconnected or broken cables can be detected or device warnings can be used on system level. In addition automatic configuration transfer can reduce time and effort if a device has to be replaced.

In automated manufacturing equipment a large number of proximity switches are used for process monitoring. Proximity switches are typically used to detect whether a target object is located within the sensing range. Sparse information on internal values and simple configuration methods like changing a detection level are available today. Ideally it is not necessary to adjust a sensor to the application or simple adjustment methods are sufficient. However, in highly dynamic applications it is sometimes not reasonable to use a standard setting or it can be challenging to find a proper application specific setting. The use of smart sensors and a proper communication interface grant access to additional information and flexible control mechanisms.

Smart Sensors

To meet the ever rising requirements in sensing distance, stability, EMC, simplicity etc. sensors are getting smarter. Configurable and programmable architectures like microcontrollers or FPGAs are used to process more and more data and to implement additional functionality. Besides the intrinsic sensing value additional accuracy information is determined and processed to achieve better system stability. Even environmental changes are taken into account. For example it is very common to implement complex temperature compensation algorithms. Internal monitoring and diagnosis algorithms are used to calculate reliability information. With more information on sensing conditions and environmental influence combined with the possibility to implement complex algorithms in small and cost sensitive devices the sensing performance is enhanced.

However, internal intelligence in smart sensors helps to achieve better sensor performance, the possibilities in configuration and the need for application specific settings arises. Therefore, smart sensors are designed for fast and simple adjustment to different situations, or one product can be used for different applications. Even configuration changes are possible to adjust a device to changing application requirements during the lifecycle.

Today, many cost sensitive smart sensors use a simple switching output to indicate the sensing condition. A standardized communication interface enables data transmission and processing of additional information. Besides monitoring of the device state additional parameters and switching signals can be calculated and transmitted. Diagnosis and process data can be used to recalibrate the process or the sensor. With the transmission of the process value continuous information is available and analog outputs can be replaced by a digital interface. Even user specific algorithms are possible in the control unit if more computing power is necessary or a complex relation between data from several sensors has to be taken into account. The values and sensor information can be read, processed and interpreted. If it is necessary the device configuration can be adjusted under operation conditions.

In many standard applications it is sufficient to use a switching output in normal operation. But sometimes it can be challenging to find a proper sensor configuration. Smart sensors provide additional information like continuous internal measurement values and reliability information for application evaluation. In combination with IO-Link the sensor runs in communication mode to optimize the sensor settings for a reliable operation. For normal operation it runs in switching mode. It is possible to toggle between the modes at any time to get additional information later on.

With the integration of a standardized digital communication interface additional information like internal measurement values, reliability information or even target properties become available. They can be used to evaluate new applications and are fundamental to find specific settings. They can also be used for process monitoring and control mechanisms. Furthermore dynamic reconfiguration methods are enabled in changing environments.

IO-Link Protocol

IO-Link is an international standard according to IEC 61131-9 as “committee draft”, developed by the IO-Link Consortium [1]. Official documentation can be found in [3], [4], [5] and [6]. The Smart Sensor Profile [4] describes a special device profile for smart sensors. IO-Link is supported by a wide range of automation companies. As one of the leading manufacturers in the automation industry ifm electronic gmbh is involved in the IO-Link Consortium [2].

IO-Link relies on standards such as M12, M8 and M5 connectors and three-wire cables and is based on point-to-point communication between the IO-Link master and one, or more, IO-Link devices. Within the automation hierarchy IO-Link provides communication to the lowest field device and acts as an extension to the superior field bus interface, see Figure 1.

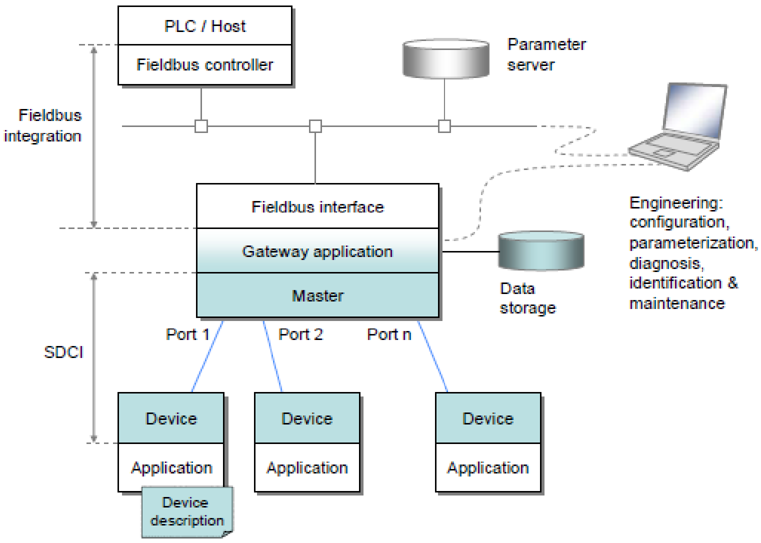

Figure 1: Domain of IO-Link (SDCI) within the automation hierarchy [3]

IO-Link establishes an extension for conventional switching signal devices, while basing on the same classic 24V physics and switching hardware. No extra wiring is necessary which causes little disturbances from a system point of view. Instead, the devices now offer two modes of operation: the classic simple switching SIO mode, and the IO-Link mode providing smart sensor features like online parameterization and diagnostics, see Figure 2.

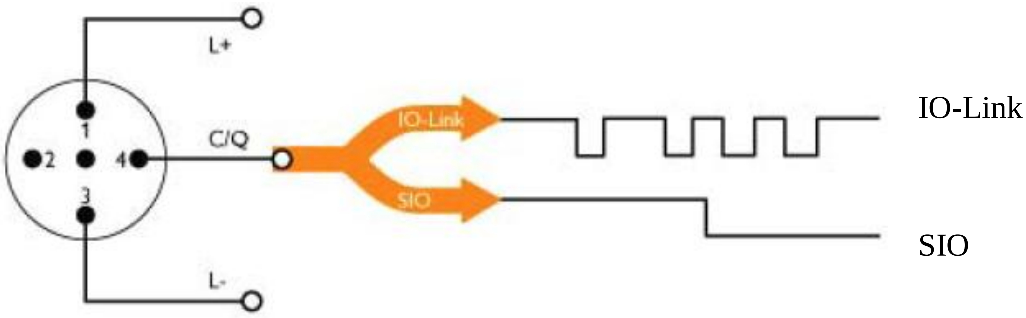

Figure 2: Wiring and two-mode operation for IO-Link devices [6]

By default, the device is in SIO mode. Upon request, the master can establish IO-Link communication at the device's specific baud rate. IO-Link features baud rates of 4.8, 38.4 and 230.4 kbit/s.
Any IO-Link data is transmitted via frames, while each frame is quantized into single-byte data chunks. Thus, IO-Link can be qualified as an 8-bit data transmission system. The crucial transmission speed qualifier is the cycle time, consisting of the frame length plus the delay times in the master and device. For a baud rate of 38.4kbit/s, the cycle time is typically 2ms.
IO-Link communication groups into various channels, basically differing in the nature of the data transmission: cyclic or acyclic. While the cyclic data incorporates the process data of the specific sensor application, the acyclic data comprises of additional communication channels for device parameterization and events and thus opens up smart sensor functionalities.

The IO-Link protocol for devices is implemented inside the IO-Link device software stack, a highly layered and modular structure consisting of finite state machines and services to exchange data and commands (refer to Figure 3). A master request is followed by data transmission from the physical layer (bottom) to the device applications (top) and back. Higher order device functions are temporally decoupled from the time-critical data layer (green) to make sure the device can respond to the master in the given response time. Each IO-Link operating is realized via multitudes of successive service operations.

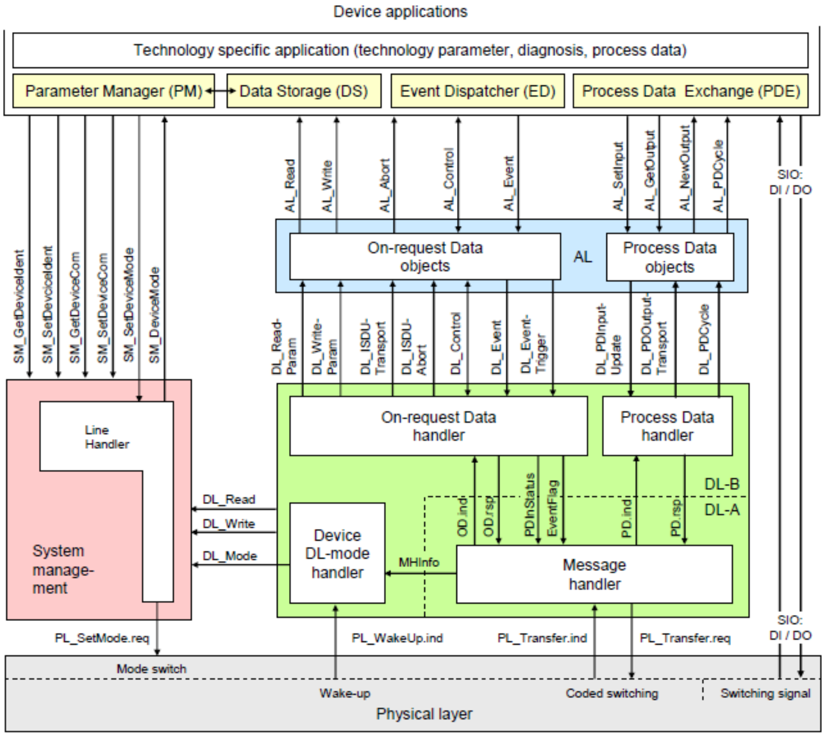

Figure 3: IO-Link device software stack – structure and services [3]

Configurable Microprocessors

We have seen that IO-Link allows to access smart sensor functionality, so that crucial functionalities become configurable during the product life cycle.

For many embedded systems availability of smart sensors has growing importance in particular because the evolution of embedded systems has taken a new direction with the appearance of Smart Products. Smart products combine features like a rich, interactive user interface, a wide range of networked connectivity, significant local processing and finally the ability to perceive the environment through a mix of sensors.

Looking at suitable processing solutions for such smart products, a new class of configurable microprocessors seems to be good choice: Hybrids of microprocessors plus modern Field-Programmable Gate-Arrays (FPGA), featuring a soft microprocessor core that is wholly implemented inside the FPGA's programmable logic as a single System-On-Chip. Examples for soft microprocessors are Nios-II from Altera and Microblaze from Xilinx. Obviously, a huge benefit now is that crucial system layers become reconfigurable: 1. the microcontroller architecture, 2. the digital signal processing (DSP) units implemented inside programmable logic and 3. the customizable peripheral set enabled by programmable I/Os.

The soft microprocessor can efficiently be released by putting the computational burden where it belongs and, thus, by utilizing the FPGA's parallel processing capabilities for DSP functions such as signal conditioning, filtering or cross-correlation functions.

With regards to the system's sensor capabilities and mixture the System-on-Chip can now on one hand host a broader spectrum of more smart sensors, even with increasing processing demands. On the other hand configurability also allows to reduce the amount of product variants by having only a few major products and generate more product variants by configuring the major products during their lifecycle.

Finally, modern FPGAs in 28nm technology offer reduced power consumption, increased system level performance and capacity and, above all, pricing options to represent a reasonable and competitive processing solution for new smart products.

IO-Link Implementation Aspects

As shown above, the IO-Link protocol architecture bases on multitudes of successive service operations via byte-oriented data types. This leads to our observation that a 32-bit ARM Cortex-M3 may not necessarily be faster that an 8-bit ATMega328. More precisely, in order to achieve higher IO-Link baud rates and smaller cycle times, switching to a microprocessor with higher data width might not be the ultimate solution. A more promising approach, already performed with IO-Link devices, is to release the microprocessor by moving the time-critical parts of the IO-Link stack – the physical layer and the data layer - out of the microprocessor.

Choosing a configurable microcontroller based on an FPGA, one can implement time-critical layers in programmable logic. Requiring only a few FPGA clock cycles per IO-Link message compared to a few hundred clock cycles in a microprocessor, this can result in a huge performance boost, while the processing burden is put where it belongs. The soft microprocessor itself, e.g. a small Nios-II core, would only have to deal with the sensor-specific applications. Or, vice versa, offloading the digital signal processing into the programmable logic frees up additional CPU cycles for handling the IO-Link protocol.

Finally, smart products demanding for a broad spectrum of sensors and significant local processing seem to greatly benefit from FPGA-based configurable microprocessor solutions.

Summary & Results

ifm electronic gmbh and Missing Link Electronics have implemented a portable IO-Link device stack which has been validated for commercial proximity switches.

Using modern software design principles such as parameterization and inheritance available in the ANSI/ISO C++ programming language we have designed a highly modular and adaptable IO-Link stack, fully compliant with the standard. This allows to rapidly, and efficiently, equip many different sensor and actuator products with IO-Link communication functionality, almost independent of the underlying microcontroller architecture.

Table 1 shows compilation results regarding the IO-Link stack footprint for a collection of common microcontroller architectures, Atmel ATMega328, ARM Cortex-M3, Altera Nios-II and Xilinx Microblaze. Results indicate a good trade-off between portability, configurability and footprint.

Microcontroller architecture	Compiler (optimization)	IO-Link stack footprint (kB)	
		Code size (kB)	Data size (kB)
Atmel ATMega328	IAR v6.50.2 (speed)	15.9	1.0
	IAR v6.50.2 (size)	14.4	1.0
	AVRGCC v4.6.2 (speed)	20.9	0.9
	AVRGCC v4.6.2 (size)	17.1	0.9
ARM Cortex-M3	IAR v6.50.2 (speed)	16.4	1.8
	IAR v6.50.2 (size)	13.6	1.8
	ARMGCC v4.6.1 (speed)	18.8	2.9
	ARMGCC v4.6.1 (size)	15.9	2.9
Altera Nios-II	Nios-II 10.1 (speed)	28	
	Nios-II 10.1 (size)	27	
Xilinx Microblaze	Xilinx SDK 14.3 (speed)	40.9	
	Xilinx SDK 14.3 (size)	34.7	

Table 1: IO-Link stack footprint for common microcontroller architectures

Authors:

Johannes Roettig received a Diploma degree in electrical engineering from the University of Ulm in 2010. At Missing Link Electronics he is responsible for business development around new applications for modern FPGA-based System-on-Chips.

Dr. Endric Schubert is a technologist in electronic system design, EDA and semiconductors. He is Cofounder of Missing Link Electronics, an embedded system company focusing on configurable system design platforms. His background includes software engineering, FPGA technology, Reconfigurable Computing and Embedded Systems design. Endric did his Diploma in Electrical Engineering at the University of Karlsruhe, Germany and his PhD in Computer Science at the University of Tuebingen, Germany. He has authored various technical publications and is inventor of several patents.

Dr. Markus Tahedl received a Diploma and doctoral degree in electrical engineering from the University of Ulm, Germany, in 2001 and 2007 respectively. As department manager at ifm electronic gmbh he is responsible for sensor, electronics and software development for position sensors. His background includes circuit and system design, sensing technologies and signal processing methods.

Bibliography:

1. IO-Link Consortium
   http://www.io-link.com/

2. ifm electronic gmbh
   http://www.ifm.com/

3. IO-Link Interface Specification
   http://www.io-link.com/share/Downloads/IOL-Interface-Spec_10002_V111_Oct11.pdf

4. IO-Link Smart Sensor Profile
   http://www.io-link.com/share/Downloads/IOL-Smart-Sensor-Profile-Spec_10042_V10_Oct11.pdf

5. IO-Link Test Specification
   http://www.io-link.com/share/Downloads/IOL-Test-Spec_10032_V11_May11.pdf

6. IO-Link System Overview
   http://www.io-link.com/share/Downloads/IO-Link_system_overview.pdf