Field Bus Systems in Motion Control – Trinamic’s View

The ability to transform digital information into physical motion is a base requirement for the 4th Industrial Revolution. This white paper highlights, discusses, and compares the most common communication (bus) systems, their typical attributes, and use cases related to the 4th Industrial Revolution.

Introduction

The ability to transform digital information into physical motion is a base requirement for the 4th Industrial Revolution. It determines the impact of cyber-physical systems, IoT, and smart technologies on our way of working and living. The 4th Industrial Revolution strongly relies on Artificial Intelligence, Image and Sensor Processing, and Automation. Every automated system – regardless if it is a small/single machine, a robot or cobot, an assembly line, a factory floor, or connected fabs – shows a similar overall structure: there is one or more controlling entity, a communication system, and the ‘endpoints’ that represent the Physical Edge. There is always a Physical Edge! Physical Edge devices can be simple I/Os, sensors, actuators, or even complex units that operate outside of the cloud at the edge of the network where real-time processing of data is required.

Sensors for visual sensing, monitoring tasks, or diagnosis tasks are required by the AI to get feedback from the Physical Edge (real world) to actually close their various feedback loops to interact with the Physical Edge and carry out their control tasks. Actuators are
typically represented by electric motor drives – inductive loads that need to be controlled with increasing demands on precision, efficiency, and synchronization (multi-axis / interaction). Examples are solenoids, DC motors, stepper motors, and servo drives. All of these are typically combined with (position) sensors as well. If sensors are considered to be the ‘eyes’ and ‘ears’ of cyber-physical systems, electrical actuators are the ‘arms’ and ‘legs’ to physically manipulate according to their respective task: close a valve, control a robot’s arm, move a belt, or open a door.

The exchange of data/information to and from the Physical Edge will gain more and more attention over the next years. All industries and industrial processes will be communication-centric, turning communication into the backbone of Cyber-physical Systems.

This white paper highlights, discusses, and compares most common communication (bus)systems, their typical attributes, and use cases related to the 4th Industrial Revolution.

**Communication Bus Systems in Motion Control Applications**

Information must be transmitted between the controlling entity and the physical end points (slaves) to read sensor data or to command motor drives to do a specific move. This is not only important in single-axis applications but is strongly required in multi-axis systems where two or more actuators must be synchronized.

In motion control systems different communication bus systems and protocols are used for this purpose. Each one with its own specific attributes and requirements. Typical bus systems in motion control applications and supported by Trinamic are RS485, CAN, and EtherCAT®.

<table>
<thead>
<tr>
<th>Bus</th>
<th>Bit Rate</th>
<th>Cable</th>
<th>Hardware</th>
<th>Max. Slaves</th>
<th>Realtime</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS485</td>
<td>9600bps...1MBit/s</td>
<td>2-wires</td>
<td>Transceiver</td>
<td>255</td>
<td>No</td>
<td>TMCL™, Modbus</td>
</tr>
<tr>
<td>CAN</td>
<td>20kBit/s...1MBit/s</td>
<td>2-wires</td>
<td>Transceiver</td>
<td>127</td>
<td>Limited</td>
<td>TMCL™, CANopen®</td>
</tr>
<tr>
<td>EtherCAT®</td>
<td>100MBit/s</td>
<td>Ethernet</td>
<td>ESC + Phys</td>
<td>65535</td>
<td>Yes</td>
<td>CoE</td>
</tr>
</tbody>
</table>

*Table 1: Comparison of RS485, CAN and EtherCAT®*

There are many parameters that can be used to compare different field bus systems. This can be the data transmission speed, the maximum number of slaves, and if the bus systems can fulfill real-time requirements. Also the hardware costs are an important parameter. RS485 has the lowest hardware costs. With CAN the hardware costs are slightly higher. With EtherCAT® hardware costs are much higher, caused by the special EtherCAT® chip (ESC = EtherCAT® slave controller) and the Ethernet Phys that are needed here.

The next sections shed some more light on these communication systems and typical application layer protocols.
RS485 Bus

Hardware Layer

The RS485 bus is the oldest field bus system discussed in this paper. It has the least hardware costs and is easy to implement. Only a UART is used, which is available on most microcontrollers, and an RS485 transceiver chip is needed, which is typically small and cheap in comparison. The cabling is also easy. Just two wires are needed for the differential signal (RS485+ and RS485-). The maximum number of nodes mainly depends on the used transceivers. In most cases this can be up to 255 nodes. The maximum bit rate is mainly limited by the overall cable length. The typical setting of the TRINAMIC Motion Control modules is 9600bps, but 115200bps or more can be used as well.

Many master interfaces cannot cope with more than 115200bps, but there are also RS485 masters that are capable of up to 1MBit/s.

High-Level Protocols

There are many different high-level communication protocols that can be used with the RS485 bus. For example there are the Proﬁbus protocol, the Modbus protocol and TMCL™, the TRINAMIC Motion Control Language. All these protocols can be used with RS485. There are also other proprietary protocols used with RS485. Most of these high-level protocols cannot co-exist on one RS485 bus, so all nodes must use the same protocol.

TRINAMIC Motion Control modules normally use TMCL™ as their RS485 protocols. For some modules, an implementation of the Modbus protocol is also available as a special firmware. These protocols do not fulﬁll real-time requirements. While this makes RS485 less suitable for applications with real-time requirements, such as cobots and highly synchronized multi-axis applications, it’s simplicity makes it the preferred bus system for motion control systems with independent motor axis or for cost-sensitive applications.

In most cases, an RS485 bus can only be used with one master. Normally a strict master-slave communication is needed, so that slaves are not entitled to initiate a communication on their own. The RS485 hardware can be extended in a way that implements a kind of collision detection (with the help of the software). This can also make it possible to entitle slaves to start a communication on their own, and multi-master systems can be made possible this way as well. But mostly this is generally not used. TRINAMIC modules also do not support this, and hence TMCL™ as an RS485 high-level protocol can only be used in a strict master-slave communication way.

TMCL™ as an RS485 high-level protocol can cope with a maximum of 255 nodes.
CAN Bus

Bus Physics and Plain CAN

A little bit like the RS485 bus, the CAN bus also uses two wires for a differential serial data transmission. The CAN bus also uses small and cheap transceiver chips, but needs a special interface that has to be implemented in the microcontroller. Such a CAN interface implements things like check sum, identifier and carries sense and collision detection in hardware. The bit rate can be up to 1MBit/s. With Trinamic modules mostly 1MBit/s is used, but also lower bit rates like 250kBit/s can be used.

The number of nodes is normally limited to 127, and the maximum bit rate that can be used is limited by the length of the bus cable. There is a rule of thumb that describes dependency: cable length <= 40000 Kbit/s * 1 m / BitRate[Kbit/s]

<table>
<thead>
<tr>
<th>Bit Rate</th>
<th>Cable Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mbit/s</td>
<td>25m</td>
</tr>
<tr>
<td>500kbit/s</td>
<td>100m</td>
</tr>
<tr>
<td>250kbit/s</td>
<td>250m</td>
</tr>
<tr>
<td>50Kbit/s</td>
<td>1000m</td>
</tr>
</tbody>
</table>

Table 2: CAN bus cable length vs. bit rate

With the CAN bus it is possible to transfer data packets of up to eight bytes each. Every packet has a built-in identifier and CRC check sum. A CRC error automatically leads to a re-transmission (this is done automatically by the hardware). This means that a safe data transfer is already implemented in hardware, so the software does not have to cope with this. The collision detection mechanism also makes the CAN bus capable of using multi-master protocols. Slaves can initiate communication on their own, so the software does not have to cope with this either.

A TMCL™ data packet fits into an eight-byte CAN data packet, so it is easy to use TMCL™ as a high-level protocol for the CAN bus. With standard firmware, this is possible with all Trinamic modules that are equipped with a CAN interface. The TMCL™ protocol used with the CAN bus can also use more than one master, and can use as much nodes as are supported by the CAN bus hardware.
CANopen®

CANopen® is the mostly used high-level protocol for the CAN bus. It has been defined by the CAN in Automation (CiA) group from the year 1992 on. The CANopen® standard defines a basic data format and protocol, using data types called objects (variables numbered using a 16-bit index and an 8-bit sub-index). Objects can be read and written using SDOs (service data objects) as well as PDOs (protocol data objects). PDOs can also be sent out by the slaves in an event-driven way. It is also possible to write objects on many slaves in a synchronized way using PDO (synchronized PDOs).

The CANopen® standard defines device profiles too. These device profiles define how CANopen® is used for different purposes. Trinamic modules use the CiA-402 profile – the CANopen® profile for motion control modules.

Most Trinamic modules equipped with the CAN interface are also offered with the CANopen® firmware.

Real-Time Capabilities

Using synchronous PDOs, some limited real-time capabilities are available with CANopen®. The master can send out PDOs to write to the data objects of the slave, and then can send out one synchronization object to command all slaves to process this data at the same time. This provides an easy to use way of synchronizing multiple modules, also under real-time requirements.

Multimaster Capabilities

Under normal circumstances, CANopen® only uses one master. But using a special extension (called ‘flying master’), more than one masters are possible. This is mainly used for marine applications.

Using the CAN bus, TMCL™ can also be used with more than one master.

CAN-FD

The CAN-FD (FD = flexible data rate) standard expands the maximum size of CAN data packets to 64 bytes instead of 8 bytes. In such data packets, the payload data is being sent using an eight times higher bit rate than the bit rate used for the identifier and the CRC. This way, such a 64 byte data packet does not need more time for the transfer than an 8 byte data packet using the normal CAN standard. The first microcontrollers that have a CAN interface that can also use CAN-FD are available now.

TMCL™ does not use the CAN-FD extensions up to now. With CANopen-FD, the CANopen® standard has been extended for using the CAN-FD capabilities. But it will take some time until CANopen-FD is being widely used. Trinamic modules do not use CANopen-FD at the moment.
EtherCAT®

Hardware Layer
EtherCAT® stands for "Ethernet in Control and Automation Technology", and thus EtherCAT® uses the standard Ethernet physical layer for the data transfer. So, standard Ethernet cables are used to connect the devices. In most cases, each slave is equipped with two Ethernet sockets: one input (link in) and one output (link out). The bus is connected from the master to the first slave, then from the first slave to the second slave and so on. When a data packet is being sent through the bus, each slaves reads data from the place assigned to it and then inserts new data at the same place.

The data transfer is done completely in hardware. This means that a special chip, called EtherCAT® Slave Controller (ESC) is needed for each slave. Examples for ESCs are the Trinamic TMC8461 and TMC8462 as well as the Beckhoff ET1100 and ET1200. Each slave also needs two Ethernet Phys. This means that compared to the other field bus systems discussed in this paper, EtherCAT® has the highest hardware costs. The TMC8462 ESC with its integrated Phys significantly reduces these hardware costs.

The advantage of this approach is that the software in an EtherCAT® slave does not have to deal with the data transfer at all. It just reads out the data from the ESC and writes new data to the ESC. This also makes fulfilling real-time requirements a lot easier. But in order to be able to completely fulfill real-time requirements, the software in an EtherCAT® slave has to process the data fast enough, so also a microcontroller that can process the data fast enough is needed.

The process data is normally being sent cyclically through the bus by the master. In most applications the cycle time is 1ms, but with some masters and slaves, also cycle times down to 250µs are possible. The EtherCAT® bus only uses one master, there is no multi-master capability.

Standardization
EtherCAT® is a highly standardized field bus system. The EtherCAT® Technology Group (ETG), founded by Beckhoff, the inventor of EtherCAT®, defines all EtherCAT® standards. The ETG also provides a conformance test for EtherCAT® slaves. Each slave that goes to the market must have passed the conformance test. The conformance test is done using a conformance test tool (CTT). This is a software product main-tained by the ETG which checks the conformance. The conformance test can be done by a slave vendor himself. Optionally, a slave can also be sent to the conformance test center of the ETG to pass an o2zicial conformance test. Once this test has been passed, the EtherCAT® logo may be printed on the slave.

For checking the interoperability of EtherCAT® masters and slaves, plug fests conducted by the ETG are held on a regular basis (at least twice a year in Europe). During these plug fests, developers of EtherCAT masters and slaves bring their devices and test them with others, ensuring secure communication between the different devices.
The rather new EtherCAT-P standard also allows to power slaves over the EtherCAT® cable. Special power modules have to be inserted into the bus which then provide electrical power to the slaves.

**CoE as a High-Level Protocol**

EtherCAT® allows to use different high-level protocols. One of these high-level protocols is CANopen® over EtherCAT® (CoE). It uses the same standards and device profiles as CANopen®, but uses EtherCAT® instead of the CAN bus as the transmitting medium. This approach merges the advantages of the CANopen® protocol with the advantages of the EtherCAT® field bus system. The advantage of the CANopen® standard is that it is being widely used in many applications, and the main advantage of EtherCAT® is its high speed and real-time capability. So CANopen® is enabled for use in a real-time environment. As such, CoE is ideal for robots and general automation in industrial, medical, and lab automation applications where high reliability, high synchronicity, and high bandwidth is required.

All Trinamic modules that are equipped with EtherCAT® interfaces support the CoE protocol standard. All EtherCAT® masters also support CoE, so Trinamic motion control modules can be used with most EtherCAT® masters without problems.

**TMCL™**

With the bus systems taking care of the infrastructure for communication between the controlling entity and the physical end points, which can be within one single device or between different devices, the communication that tells the actuators what to do is still missing. For this, communication protocols are needed. Trinamic modules usually support two protocols: CANopen and TMCL™.

TMCL, the TRINAMIC Motion Control Language, can be used for two purposes: First, it can be used as a communication protocol for motion control modules. Second, it can be used as a programming language for scripts that are being downloaded to the motion control modules and executed there.

**Summary**

Embedded systems are used in various industries and applications for years now, making work – and increasingly everyday life – easier by transforming digital information to physical motion. Companies are now focusing their efforts toward further integration of these systems with sensors and copious amounts of data. By doing so, they create cyber-physical systems by merging pervasive computing and information with hardware and software, which in turn enables predictive maintenance, increases speed and intelligence, and improves the user experience.

This trend towards further integration that’s taking place in the 4th Industrial Revolution calls for a fast, secure, and reliable link between master and slaves, creating a robust bridge between the cloud and the physical edge. And while this takes place both within one single device and between different devices or machines, the paradigm is shifting
towards the communication between devices, or machine-to-machine communication (M2M). Not only between devices used on the factory floor, also lab automation, office automation, and even homes see an increase in M2M communication. While M2M communication at home mostly takes place via a wireless network, professional environments will always rely on bus systems for secure, real-time communication.

However, different real-time requirements demand different approaches. As a result, multiple bus systems are likely to stay in use for various reasons: First of all, not every system has the same requirements when it comes to one or multiple masters, latency demands, etc. Second, protocols can be restricted to a specific bus system, which could make it more attractive to companies to go for one bus system even if there's a (slightly) better alternative. It is even to be expected that the offer will expand as the next generation of integrated systems come with their own, unique requirements such as increased precision, speed, and synchronicity, that are not supported yet by existing bus systems.

About TRINAMIC Motion Control

Instead of semiconductor makers that just want to sell more chips, TRINAMIC Motion Control (TMC) began with engineers who wanted to design better drives. Its founders knew that designing motion control technology in hardware, not software, was essential for motors that work efficiently, reliably, smoothly, and quietly.

The Hamburg, Germany-based fabless semiconductor company's own motion control architecture divides power electronics into three building blocks: a microcontroller for digital processing, a driver that translates digital signals into electrical power, and a gate driver plus power switches. This simplifies development, and makes it easier to design components best suited for each block. In 2015, TMC became the first supplier worldwide with an EtherCAT-compliant slave-controller IC for integrating complex, real-time, latency-free I/O peripherals.

TMC now has over 20 years of application experience in embedded motion control. It develops and sells ICs, modules, and integrated mechatronics to market-leading manufacturers worldwide. Examples include Ecovent, GE, Honeywell, Kuka, Markforged, Össur, Xerox, and Zeiss. The company's products have been integrated into a wide range of applications in digital manufacturing, biotechnology, lab automation, materials handling, and factory automation.