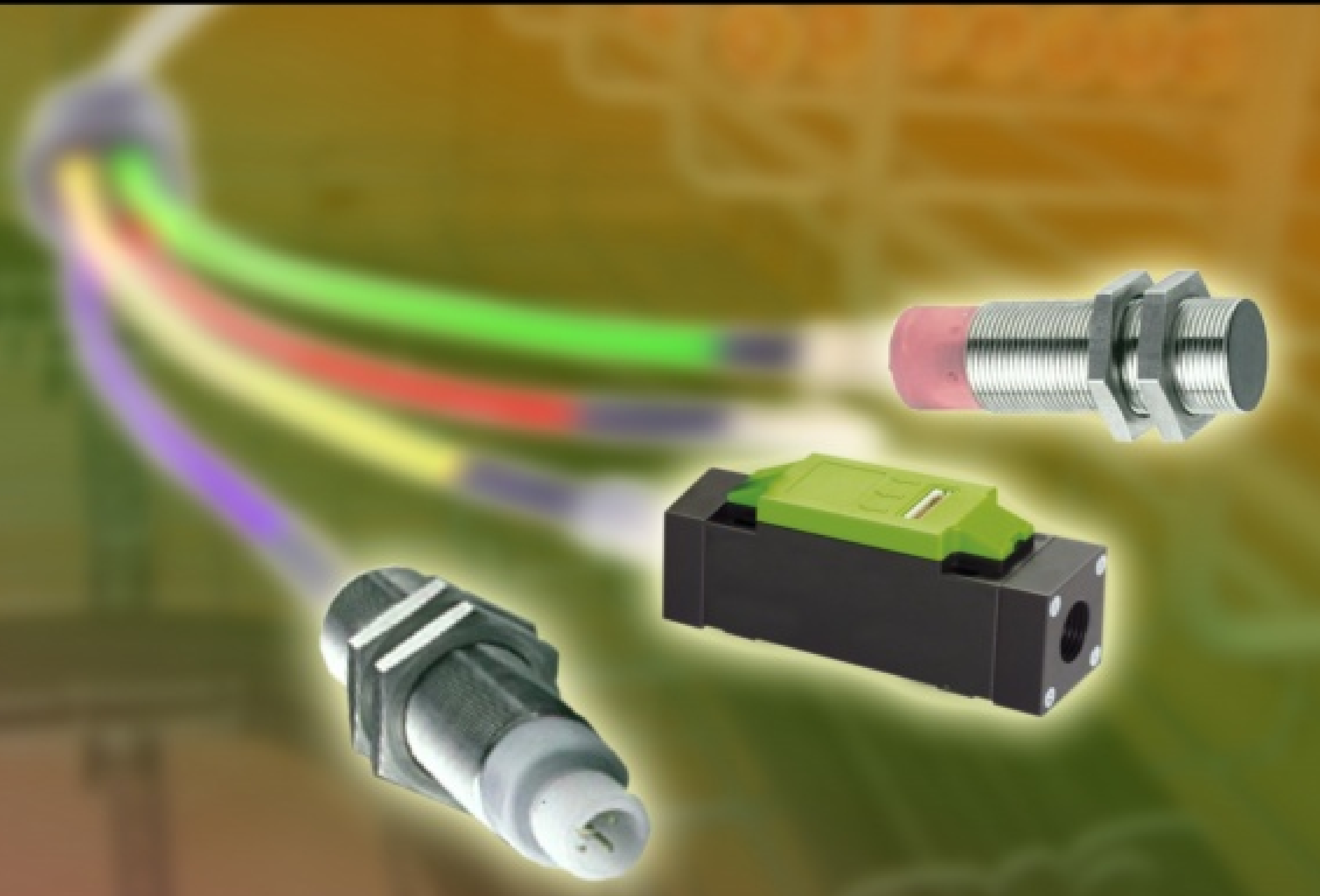
ISSN 1726-5749

SENSORS & TRANSDUCERS

5/07 vol. 79



Sensor Buses and Interfaces

International Frequency Sensor Association Publishing

IFSA

Sensors & Transducers

Volume 79
Issue 5
May 2007

www.sensorsportal.com

ISSN 1726-5479

Sensors & Transducers Journal (ISSN 1726-5479) is a peer review international journal published monthly online by International Frequency Sensor Association (IFSA). Available in electronic and CD-ROM.

Sensors & Transducers Journal

Contents

Volume 79
Issue 5
May 2007

www.sensorsportal.com

ISSN 1726-5479

Research Articles

Authors are encouraged to submit article in MS Word (doc) and Acrobat (pdf) formats by e-mail: editor@sensorsportal.com
Please visit journal's webpage with preparation instructions: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm

Sensors & Transducers Journal, Vol.79, Issue 5, May 2007, pp. 1143-1150

Sensors & Transducers

ISSN 1726-5479

http://www.sensorsportal.com

Standardized Interconnectivity of Sensors for Construction Machines via CAN Bus with the Higher-Layer Protocol CANopen

Christian DRESSLER
CAN in Automation (CiA), Am Weichselgarten 26, DE-91058 Erlangen, Germany
E-mail: headquarters@can-cia.org, http://www.can-cia.org

Received: 1 February 2007 /Accepted: 11 May 2007 /Published: 31 May 2007

Abstract: The paper provides the functionality of CANopen and introduces the specifications, which define the integration of sensors in construction machines. The paper will finish with a detailed description of an example network configuration. *Copyright © 2007 IFSA.*

Keywords: Bus, Network, CANopen, Mobile, Construction

---

1. Introduction

When the European research project OSYRIS was looking for an appropriate sensor communication standard, the participating companies decided to chose CANopen. The requirements were plug-and-play capability, dedicated sensors and the specification of set points for the sensors.

The specification CiA 415, the "CANopen device profile for road construction machinery" was jointly developed within the OSYRIS project (Open System for Road Information Support) and supported by the European Asphalt Pavement Association (EAPA).

2. Fundamentals

The components for mobile usage have to prevail under extreme conditions and be easy to integrate and to be operated. There exist a lot of different types of construction machines on the market. But as different machines often have the same types of sensors installed, the amount of different sensor-types used on the machines is not as high. An overview of the sensor-types which have been specified in CiA 415 is shown in Table 1.

The requirements to the sensor devices are very high: Due to the rough conditions like extreme temperatures, very high humidity, vibrations or strokes, appropriate connectors, cables and packaging has to be used. Because in mobile machines the environment can't be controlled, the immunity to EMI is also an important factor. A sensor system for a construction machine should be quick and easy installed. This assumes that hardware and devices do already exist on the market. Existing software for development or diagnostics will also reduce the time-to-market. The sensors should have plug-and-play capability and the network should be able to be split to an integration network. This allows different bit-rates, longer distances and improved redundancy.

Table 1. Sensor Types supported by CiA 415.

Sensor types	
3D absolute position	Transversal roughness
Angle Position	Tool extend
Curvilinear coordinates	Material volume
Curvilinear angle position	Material core temperature
Geographical coordinates	Mass flow
Level-deviation	Volume flow
Horizontal deviation	Ambient temperature
Steering angles	Base temperature
Machine speed	Wind speed
Traveled distance	Surface temperature
Rotational speed	Wind direction
Layer measurements	Humidity
Longitudinal roughness	Barometric pressure
Water tank level	Rainfall

Standardization has advantages for the sensor manufacturer. He doesn't need to customize his product to the requirements of each client every time. He gets the chance to compete with other OEMs. On the other hand, the system integrator is interested in huge choice of products which work together in the machine without customizing them first. The availability of standardized products on the market reduces the time-to-market of the machine.

CAN is standardized in ISO 11898-1 to 3, the CANopen profile CiA 301 "Application Layer and Communication Profile Version 4.0" is described by EN 50325-4.

CAN is a serial bus system, which was originally developed for automotive applications in the early 1980's. CAN, which is by now available from around 40 semiconductor manufacturers in hardware, provides two communication services: the sending of a message and the requesting of a message. All other services such as error signaling, automatic re-transmission of erroneous frames are user-transparent, which means the CAN chip automatically performs these services.

The voltage of the differential signal is according to ISO 11898-2 between -2 and +7 Volt which provides great immunity to EMI. The nominal specific propagation delay of the two-wire bus line is specified at 5 ns/m. CiA 102 defines the bit-timing for rates of 10 Kbit/s to 1 Mbit/s. The data is protected by the data link layer very well (worst case residual error probability of $P = 7 \cdot 10^{-9}$).

3. CANopen Basics

Since the CAN standard just describes the transmission of data over the bus, CANopen describes how to interpret the content of those data messages. According to the OSI Reference Model, CAN provides layer 1 (Physical Layer) and layer 2 (Data Link Layer), whereas CANopen acts as layer 7 (Application Layer). CANopen is a widely used and flexible standard which can be adapted to many different applications.

The CANopen Object Dictionary:

The Object Dictionary (OD) is a data storage which has to be implemented in the application of every node. It is a grouping of objects stored in a look-up table. The OD is the interface between the communication on the bus and the software application of the nodes. It is organized as index, so every object has a unique index-number and can additionally be divided into sub-index-numbers.

An object consists of the object-name, the object-code (e.g. variable, array), the data type (e.g. unsigned16, integer8), the entry category (if implementation of the object is mandatory or optional), the access type (e.g. read-only, read-write), the value range and a default value. The OD stores data such as process data, identification properties or configuration and communication parameters.

CANopen communication model:

CANopen defines two types of transmission services. They are called Process Data Object (PDO) service and Service Data Object (SDO) service. Additionally, it provides a Network Management (NMT) service, Node monitoring services and an emergency object (EMCY), which indicates errors quickly.

PDO: The CAN message consists besides some status flags just of a unique identifier and the data field. A PDO is a data message with no overhead data, the whole space of the data field can be used to transmit the process data, no CANopen specific control-bits are sent. The size of the process data per message is limited to 8 byte, due to ISO 11898-1. The message is broadcasted (producer/consumer-model) to all nodes and is unconfirmed. So it can be received by zero or more nodes, depending on whether the connected nodes are accepting the message-identifier or not. The transmission type describes the kind of trigger to send a PDO. A message can be sent if a timer runs out, if a specific value changed, or if the so-called sync-signal is received. A PDO which is sent to another node is called Transmit PDO (TPDO). A TPDO which is received from another node is called Receive-PDO (RPDO). To configure PDO communication, the PDOs have to be mapped: The PDO mapping assigns the content of an object to a message-identifier. This applies for TPDOs as for RPDOs. A TPDO reads out the mapped object before sending its content, the content of an RPDO is written into an object according to the mapping in the OD. The mapping in the OD can be configured whether via SDO or can already be pre-defined. The PDO service is normally used to submit data under real-time conditions with no need for confirmation.

SDO: The SDO protocol defines the communication between two nodes (server/client-model). The service is confirmed and the amount of the data to be transmitted is not limited. If the size of the data which shall be transmitted exceeds the maximum available space, the data is split into many messages. As the service is confirmed, two CAN-IDs are required for the communication between server and client. One CAN-ID for messages to be received by the server and one ID for those messages intended to be received by the client. This service is normally used to configure a network: The configuring node is a client, whereas the rest of the nodes are servers. SDOs are used for example for: setting device parameters, assigning application data to PDOs or assigning message identifiers.

Figure 1 depicts the interface functionality of CANopen. The raw data of a sensor device, e. g. voltage, is sent to an application. This software is the manufacturer-specific interpretation of the requirements, described by a CANopen profile. In the next step the calculated values are assigned to the right indices of the Object Dictionary as defined in the CANopen profile. To enable read- and write-access from other devices to the Object Dictionary, the protocol stack defines the communication protocols.

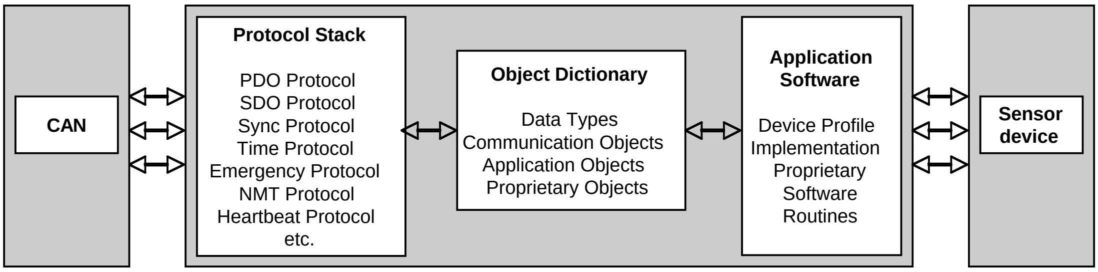


Fig. 1. Between sensor device and CAN, the CANopen application manages the communication.

4. CANopen Documents for Sensor Integration

Since the specification CiA 301 just defines the basics of communication, CiA 415 defines each sensor of a whole sensor application. To use the sensor application profile in an integrated network, the application profile CiA 436 has been developed. It defines the communication between sub-networks, such as sensors, user interface, engine or transmission (Figure 2).

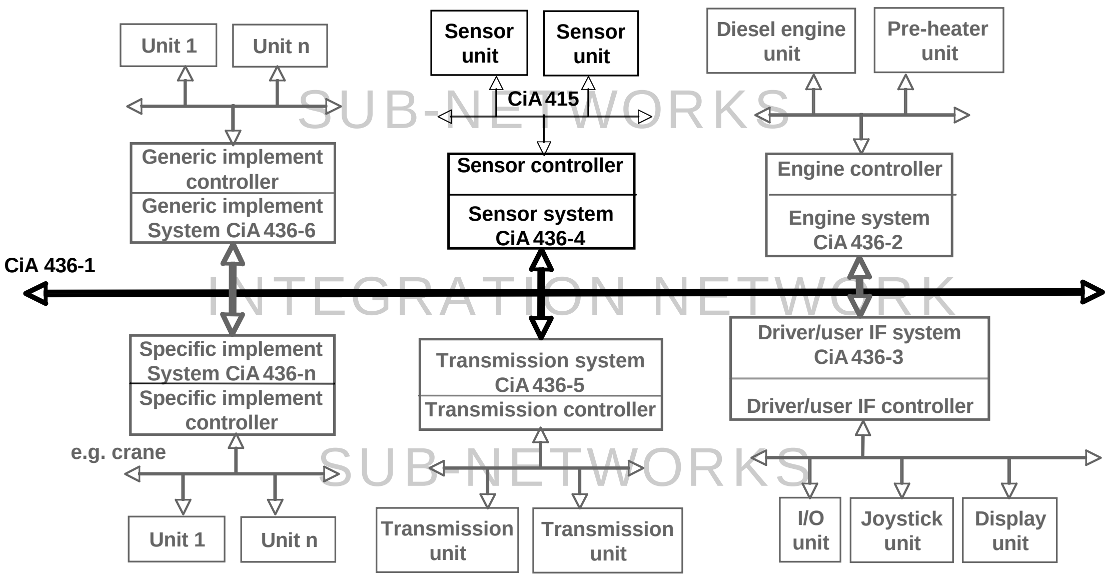


Fig. 2. The structure of the integration network according to CiA 436 including CiA 415.

CiA 301: This specification provides services like SDO, PDO, EMCY, Heartbeat or NMT. CiA 301 has to be implemented in every device, which shall be conforming according to CANopen.

CiA 415: This specification defines extra services, connectors and the data provided by the sensors. To provide auto-configuration of the sensor devices, a master-slave structure is required. The sensor controller (master) provides the CANopen manager functions. The specification defines a special boot-up process as shown in Figure 3: The sensor controller scans all 127 node-IDs available in one network. The request to the sensor devices is done via the SDO service. If the access to object 0x1000 (device type) of a node does not fail, the object 0x6010 of the object dictionary is read. The amount of provided sensor process data can be read from sub-index 0x00. The following sub-indices are a list of supported objects. Such objects provide process data such as the temperature of the processed layer or the wind speed. With this information, the TPDO mapping and communication parameters for the sensor devices are generated and downloaded to them. Thus, the sensor controller has determined which device receives which messages. The control unit also knows now, how many physical devices according to CiA 415 are available in the network and how many virtual devices (functionalities) each physical device includes. The next information the controller unit is interested in, is the kind of data each sensor wants to receive. This information is stored in object 0x6020 of each sensor device. After successful receiving this data, the control unit can generate the RPDO parameter for the sensor devices and download them to the devices. Now, every sensor has selected, which of the other sensors' TPDOs will be received. After this process, the recognized nodes are being started (set to NMT state "Operational"). A "Data Status Request" is sent via PDO, and when no error occurs, the sensor application master will be started.

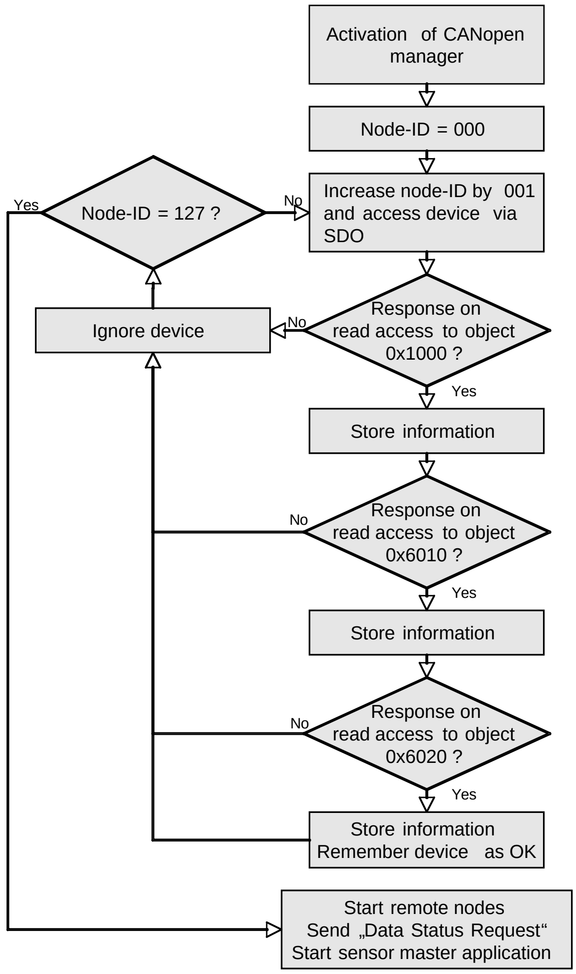


Fig. 3. Boot-up procedure of the CANopen manager.

The object dictionary begins with machine- and tool-specific objects and objects for the event transfer system are defined. Then, the sensor objects are divided into four parts. Each of the four parts contains the same list of sensor names, but with different indices and access types. These four parts are called: "Provided process data" with access type read-only, "Required process data" (read-write), "Configured reference-values (set-points)" (read-only) and "Requested reference-values (set-points)" (read-write). A sensor device needs those objects as follows: The data provided by an e.g. ambient temperature sensor is stored in the object "ambient temperature" shown in part one. If this sensor wants to receive data from the sensor "wind speed", the object "wind speed" listed in part two has to be implemented in the ambient temperature sensor. The communication directions of this example can be seen in Figure 4. The set-point of the sensor is stored in the object "ambient temperature" of part three, and if the set-point of the "surface temperature" sensor shall be received from ambient temperature sensor, object "surface temperature" from part four has to be implemented.

CiA 436: This specification defines special gateways, which are connected between a sub-network and the integration network. Real-time data can be mapped from the sub-network to the integration network. At the moment, CiA 436 consists of five parts: The first part defines common objects. They apply to all gateway devices, which are specified in the parts 2 to 6. The parts describe the sub-networks as shown in Figure 2.

5. An Example Configuration

The theory of the preceding clauses shall now be demonstrated with the aid of a pictorial example. This is not an example taken from the practical use. In our imagination, the sensor application providing the ambient temperature around the machine wants to adjust the temperature values in real-time. The sensor shall receive the wind speed and the surface temperature directly to take their values into account of its calculation. In Figure 4 a flowchart is shown, which describes the data transfer of the devices in the sensor network. Each of the three sensor devices submits its process data to the master. Additionally, the sensor application which is processing the ambient temperature shall receive the process data of the other two sensors. As we are dealing with a bus system, Figure 4 doesn't show the physical connections. In our example, the manufacturer of sensor 1 and sensor 2 has decided to integrate them into one physical device. The physical realization is shown in Figure 5. The three devices are connected to a bus. When a device is sending a message, every other device will select, if it wants to receive this message. This selection is done by the CAN-IDs. If Sensor 2 is sending a TPDO with the CAN-ID being 0x190, all devices which have RPDOs listening on CAN-ID 0x190 configured, will receive this message. In our example, the master and sensor 4 in Figure 5 will receive the wind speed as shown in Figure 4. Thus, the communication can be described with RPDOs and TPDOs.

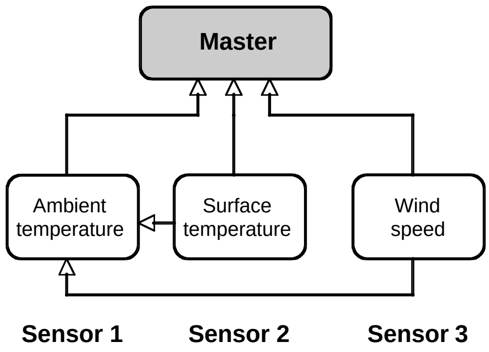


Fig. 4. The arrows describe the desired communication between the sensors and the master.

As Sensor 1 and Sensor 2 have been merged to sensor 4, they share the same micro-controller and thus the CAN-connection, the object dictionary and the application. To transmit the surface temperature to sensor 1 like shown in Figure 4, no RPDOs or TPDOs are required. The application has already access to the values. But when transmitting process data to the bus, each virtual sensor uses its own CAN-ID, so other sensor devices don't have to bother about the physical constitution of the devices in the network.

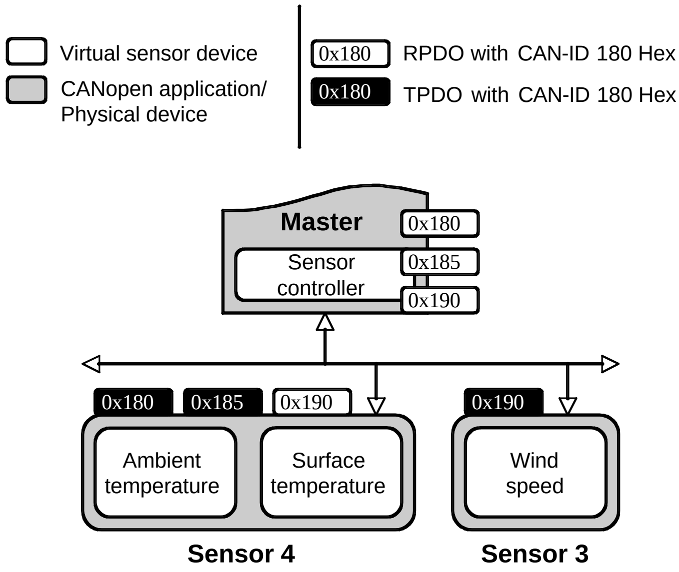


Fig. 5. An example how a sensor network can be realized with CANopen, showing the functionality of virtual devices.

To work over long distances, with different baud rates or with a limited band-with, it is recommended to split communication into own networks. To connect these networks in a standardized way, an integration network is required as shown in Figure 2.

Our example sensor 3, which is indicating the wind speed shall now be alone in the network according to CiA 415, connected to the integration network. The interface will be CiA 436-4. The second network, connected to the integration network shall include devices which have to be represented by two virtual interfaces, CiA 436-3 and CiA 436-2 as shown in Figure 6. As CiA 436 is an application profile, such as CiA 415, it also deals with virtual devices. Each interface connected to the integration network is a virtual device and thus can be merged.

If the current wind speed shall be shown on the display, the values can be read out either by access via SDO or by waiting for the appropriate PDO. When the sensor controller has received the values, it provides them to the object dictionary of the interface device according to CiA 436-4 at index 0x6542. An excerpt of each object dictionary is shown in Figure 6. The communication over the integration network to network 2 can happen on both ways, via SDO or PDO. The values are now stored in object 0x63AA of the controller's object dictionary. Now the software application has access to the values and can send them to the display. This communication can be manufacturer-specific. Due to this, the object dictionary doesn't contain any display-specific objects. Note that the indices of the integration network controllers are different to those of the sub-networks. This is because no overlapping of indices is allowed, when all parts of CiA 436 would be merged into one single object dictionary.

To enter a desired engine speed via HMI, part 3 describes object 0x64D0 where this entered value can be provided to the integration network. As part 2 and part 3 are now physically combined, these objects are not required any more for the communication between HMI and engine. But as the HMI may not be the only way to change the engine speed, the objects are still useful.

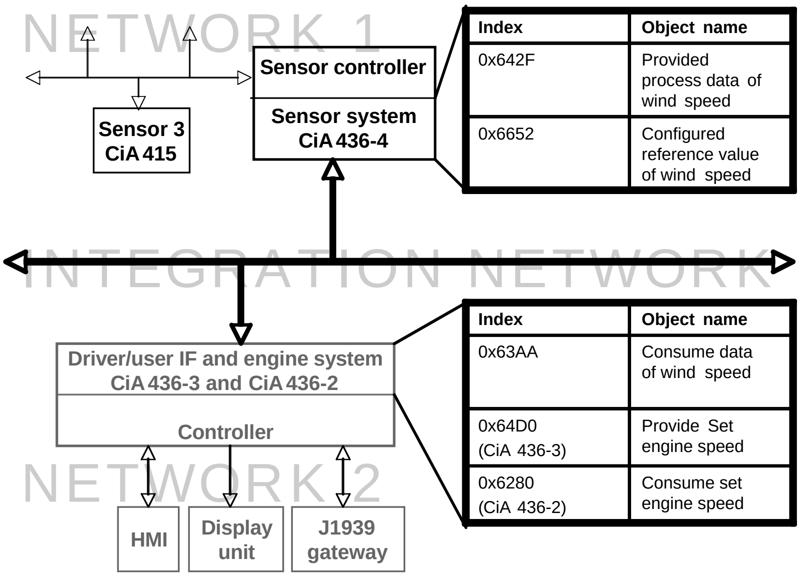


Fig. 6. An example constellation of sub-networks and integration network.

6. Conclusion

It is not easy to meet all the demands when integrating sensors on construction machines. But the car-proven robustness of CAN in combination with the customizable CANopen, which provides a special standard for a sensor system, is able to solve this task in a proper way:
The robust cables (CAN 5-wire cable including power supply), the EMI immunity due to the CAN differential balanced line, the quick and easy sensor integration and installation because of standardized components according to CiA 415 are important factors. Also the possibility of distributed sub-networks with standardized integration network according to CiA 436, the hardware availability (CAN and CANopen are wide spread) and last but not least the already existing proven software and diagnostic-tools, are part of the required features.

Guide for Contributors

Aims and Scope

Sensors & Transducers Journal (ISSN 1726- 5479) provides an advanced forum for the science and technology of physical, chemical sensors and biosensors. It publishes state-of-the-art reviews, regular research and application specific papers, short notes, letters to Editor and sensors related books reviews as well as academic, practical and commercial information of interest to its readership. Because it is an open access, peer review international journal, papers rapidly published in *Sensors & Transducers Journal* will receive a very high publicity. The journal is published monthly as twelve issues per annual by International Frequency Association (IFSA). In additional, some special sponsored and conference issues published annually.

Topics Covered

Contributions are invited on all aspects of research, development and application of the science and technology of sensors, transducers and sensor instrumentations. Topics include, but are not restricted to:

- Physical, chemical and biosensors;
- Digital, frequency, period, duty-cycle, time interval, PWM, pulse number output sensors and transducers;
- Theory, principles, effects, design, standardization and modeling;
- Smart sensors and systems;
- Sensor instrumentation;
- Virtual instruments;
- Sensors interfaces, buses and networks;
- Signal processing;
- Frequency (period, duty-cycle)-to-digital converters, ADC;
- Technologies and materials;
- Nanosensors;
- Microsystems;
- Applications.

Submission of papers

Articles should be written in English. Authors are invited to submit by e-mail editor@sensorsportal.com 4-12 pages article (including abstract, illustrations (color or grayscale), photos and references) in both: MS Word (doc) and Acrobat (pdf) formats. Detailed preparation instructions, paper example and template of manuscript are available from the journal's webpage: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm Authors must follow the instructions strictly when submitting their manuscripts.

Advertising Information

Advertising orders and enquires may be sent to sales@sensorsportal.com Please download also our media kit: http://www.sensorsportal.com/DOWNLOADS/Media_Kit_2007.PDF