CANopen Protocol for High Precision Ground Station Antenna Positioning

Sergiu CANDRAMAN
Technical University of Moldova
candraman.sergiu@gmail.com

***Abstract* — The paper deal with high precision ground station antenna control and positioning using CANopen protocol. CANopen is designed for motion-oriented machine control networks, such as handling systems. The diagram of the antenna control system based on CANopen protocol is presented.**

***Index Terms* — CANopen, CAN bus, microcontroller, network, protocol, encoder.**

I. INTRODUCTION

CANopen is a communication protocol and device profile specification for embedded systems used in automation. In terms of the OSI model, CANopen implements the layers above and including the network layer. The CANopen standard consists of an addressing scheme, several small communication protocols and an application layer defined by a device profile. The communication protocols have support for network management, device monitoring and communication between nodes, including a simple transport layer for message segmentation/desegmentation. The lower level protocol implementing the data link and physical layers is usually Controller Area Network (CAN), although devices using some other means of communication (such as Ethernet Powerlink, EtherCAT). In the paper is proposed the implementation of the CANopen protocol for ground station antenna control.

VI. CANOPEN PROTOCOL

The basic CANopen device and communication profiles are given in the CiA 301 specification released by CAN in Automation. Profiles for more specialized devices are built on top of this basic profile, and are specified in numerous other standards released by CAN in Automation, such as CiA 401 for I/O-modules and CiA 402 for motion control.

Every CANopen device has to implement certain standard features in its controlling software.

- A **communication unit** implements the protocols for messaging with the other nodes in the network
- Starting and resetting the device is controlled via a state machine. It must contain the states Initialization, Pre-operational, Operational and Stopped. The transitions between states are made by issuing a network management (NMT) communication object to the device.
- The **object dictionary** is an array of variables with a 16-bit index. Additionally, each variable can have an 8-bit subindex. The variables can be used to configure the device and reflect its environment, i.e. contain measurement data.
- The **application** part of the device actually performs the desired function of the device, after the state machine is set to the operational state. The application is configured by variables in the object dictionary and the data are sent and received through the communication layer.

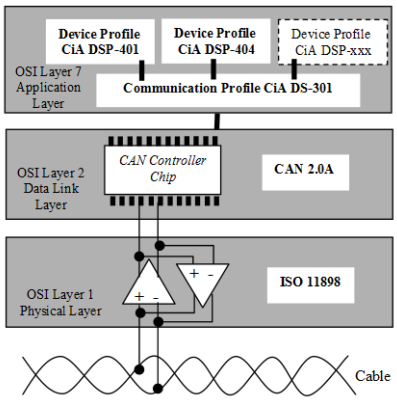

Fig.1. Schematic overview of CAN and CANopen standards in the OSI network model**.**

CANopen devices must have an object dictionary, which is used for configuration and communication with the device. An entry in the object dictionary is defined by:

- **Index**, the 16-bit address of the object in the dictionary
- **Object name**, a symbolic type of the object in the entry, such as an array, record, or simple variable
- **Name**, a string describing the entry
- **Type**, gives the datatype of the variable (or the datatype of all variables of an array)
- **Attribute**, which gives information on the access rights for this entry, this can be read/write, read-only or write-only
- The **Mandatory/Optional** field (M/O) defines whether a device conforming to the device specification has to implement this object or not

Communication between the network users and the Master (PC / Control) takes place by means of object directories and objects. The objects are addressed via a 16 bit index. The CANopen communication profile DS 301 standardizes the various communication objects.

CANopen message structure

- The first part of a message is the COB ID (Identifier).

Function code	Node ID
4-bit function code	7-bit node ID

Fig.2. CAN 2.0A ID.

A CAN telegram is made up of the COB ID and up to 8 bytes of data:

COB ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Xxx	x	xx	xx	xx	xx	xx	xx	xx	xx

Fig.3. CAN data format.

CANopen, however, is facing a much tougher battle, since its major application range is now being attacked by the new Ethernet technologies. These CANopen legacy applications are:

- Motion Control
- Industrial Machine Control

VII. CANOPEN AT GROUND STATION ANTENNA

To achieve very high precision data acquisition such as antenna positioning data from 18-bit encoders need to use one of the methods of high speed data transfer, with a hardware error correction and is referred to as priority based bus arbitration. This is the CAN network (Controlled Array Network), a communication system to transfer data in environments with potential high electromagnetic fluctuations.

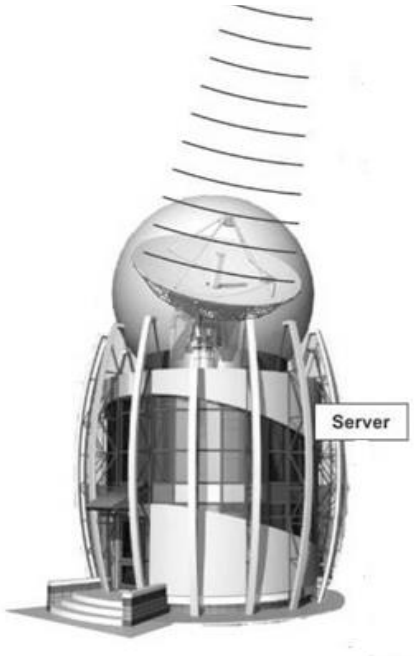

Fig.4. Ground station at Technical University of Moldova

We proposed this model for data transfer between the encoder and the main controller to make the whole system in an efficient way, such as connecting both encoders to CAN transceiver, meaning connect them in a network where we have a Master and 2 Slave 's as shown in ***Fig. 5***.

Each of slave download their data automatically in CAN registers, meaning to take less time to process data from encoders using the interrupt method and priorities allocation based on the identification code of each encoder.

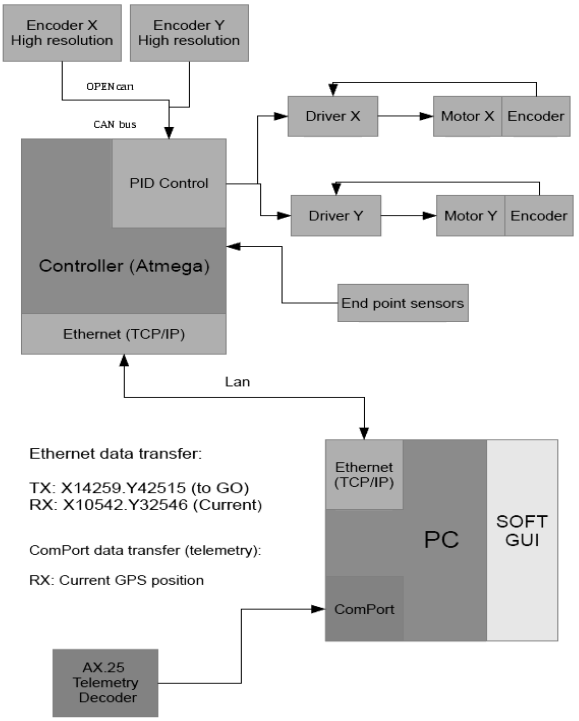

Fig.5. Diagram of the antenna control system.

VIII. CONCLUSION

Standardized profiles (device, interface and application profiles) developed by CiA members simplified for us the system design, so we can easily integrate a lot of sensors on the same bus, using identification numbers and PID's. Off-the-shelf devices, tools, and protocol stacks are widely available at reasonable prices. For system designers, it is very important to reuse application software. This requires not only communication compatibility, but also interoperability and interchangeability of devices. Using CANopen protocol and AT90CAN128 microcontroller it was realized a working CAN network based on two high resolution Encoders and one microcontroller at speed of 1 MBPS.

REFERENCES

[1] Arduino, open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Avalaible: http://arduino.cc

[2] H.Boterenbrood, CANopen high-level protocol for CAN-bus. NIKHEF, Amsterdam March 20,2000.

[3] CANopen fieldbud manual, MDrive Motion Control Products, Shrinder electric.

[4] Mike Blenderman, CAN ON THE AVR.

[5] Wilfried Voss, The future of CAN/CANopen and the Industrial Ethernet Challenge.