MicroCANopen Distributed Control Node Ethernet HTTP Monitoring

Petr Krist

Department of Applied Electronics and Telecommunications
University of West Bohemia in Pilsen – Faculty of Electrical Engineering
Univerzitni 8, 306 14 Plzen, Czech Republic
krist@kae.zcu.cz

Abstract **–The paper outlines the structure of the advanced distributed control node and describes its implementation example on the STR912 ARM platform. The node features two communication interfaces - CAN (Controller Area Network) and Ethernet, equipped with high-layer protocols MicroCANopen and HTTP web server over TCP/IP. Monitoring of application process data objects PDO of the MicroCANopen protocol using remote www client is described and the importance of the industrial Ethernet communication is discussed.**

I. INTRODUCTION

The rapid progress of the silicon and semiconductor technologies and the microelectronic systems has brought about huge deployment of the microcomputers and microcontrollers widely spread all over the world and implemented in many application areas. Formerly used local and complex stand alone central control systems have been replaced by sophisticated structures of distributed control systems consisting of smart control application nodes communicating each other in order to exchange their local application data. Nowadays it is impossible to imagine an industrial control system without a communication capability and, therefore, the industrial communications play the key role in the design and an implementation of any industrial control system.

The implementation example of distributed control node, described below, is based on the platform ARM STR912 and it features two communication interfaces – CAN and Ethernet with corresponding upper protocol layers. Prior the function and structure description of the node the main basics are mentioned.

II. CAN - CONTROLLER AREA NETWORK

CAN - specification was originally developed by Bosch Company for automotive purposes and due its features it quickly expanded into all industrial application areas. The physical layer of the high-speed version CAN ISO 11898 enables differential data transfer at bit rates up to 1 Mbit/s on a distance up to 40 meters with the characteristic impedance of the twisted-pair cable 120 Ω. Differential signal distribution method is convenient from the viewpoint of electromagnetic susceptibility in a harsh industrial environment. In addition to the physical layer, CAN standard specifies the link layer, implementing non-destructive CSMA/CR - *Carrier Sense Multiple Access/Collision Resolution* access method, based on multiple bit arbitration (Fig. 1).

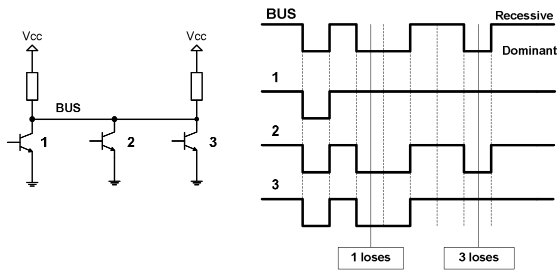

Fig.1 CAN CSMA/CR access method

The CAN frame transmission starts by 11-bit identifier *(CAN 2.0A Standard)* or 29-bit identifier *(CAN 2.0B Extended)*. The value of CAN identifier determines frame priority (the less identifier binary value the higher priority) and its value indicates the network object number (2^{11} or 2^{29}). The following parts of the CAN frame are RTR control field, data length field (up to 8 bytes can be transmitted), acknowledge field and CRC checksum.

Due its simplicity and unique features, the CAN communication is (and likely will be in the future) widespread all over the world and CAN controllers are included almost in every medium or high-class microcontroller.

In order to meet interoperability requirements, CAN application layer protocols have to be implemented. These are Device Net, SDS, CAN Kingdom, SAE 1939, and especially CANopen - the most commonly used standard in the Europe region. The importance of the CANopen underlines the fact, that this specification was adopted by the most famous industrial Ethernet solution - ETHERNET Powerlink standard.

III. CANOPEN, MICROCANOPEN

The CANopen specification defines the comfort application protocol layer for distributed industrial automation systems based on CAN. The heart of the CANopen protocol is *Object Directory* definition - **OD.** It contains general configuration, communication, and application data accessible via predefined service set *Service Data Objects* - **SDO**. Real-time sensitive technological data can be mapped into the *Process Data Objects* - **PDO** to manage fast real-time communication responses. The protocol defines by default 4 transmit TPDO and 4 receive RPDO objects of 8 bytes length, identified by their unique ID. The protocol *Network Management* - **NMT** provides the services for node configuring, monitoring, and overhead. The CANopen communication profiles (CiA/DS 301 and CiA/DS 302) and device profiles (CiA 4xx) are defined by CiA group.

Nevertheless, there are sometimes circumstances when the complete CANopen protocol specification implementation is not possible and desirable. Therefore, the light version of the CANopen specification - MicroCANopen was specified. MicroCANopen is targeted at embedded networking applications and allows the base CANopen protocol functionality either on small hardware systems (8-bit microcontrollers with CAN interface – e.g. 8051 derivatives) or on the more complex systems, where the full CANopen implementation is not required and wouldn't be employed. A communication node featuring MicroCANopen acts as the any other equivalent node in the CANopen network.

IV. ETHERNET

The most commonly used standard of the data information network was developed in the beginning of the seventies by Xerox Company and later extended in cooperation with Digital and Intel companies. The standard was accepted by IEEE in 1995 under project specification IEEE 802.3 defining physical and link layers OSI-ISO. There are various versions of the standard available nowadays – base 10 Mbit/s bit rate version, most frequently used **Fast Ethernet** featuring 100 Mbit/s bit rate, Gigabit Ethernet version 1 Gbit/s and optionally 10 Gbit/s bit rate version. The shared media access is executed due to **CSMA/CD** *(Carrier Sense Multiple Access/ Collision Detection)* access method managed by MAC link sub-layer. The MAC layer defines the Ethernet frame illustrated on the top of Fig.2. Commonly used frame type at present is the *Ethernet II* .

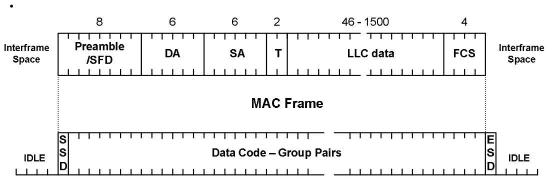

Fig.2 Fast Ethernet MAC frame

The Ethernet specification defines various physical layers. Nowadays there are two best suitable and widespread physical layers for industrial applications. These use two twisted pairs of wires:

IEEE 802.3i, referred to as **10Base-T** at the bit rate 10 Mbit/s, Manchester coding, two independent UTP or STP twisted pairs, category from 3 to 5, star network topology using active Ethernet hubs on maximal segment length of 100m. The characteristic impedance of the twisted-pair cable is 100 Ω.

IEEE 802.3u, Fast Ethernet referred to as **100Base-TX** at the bit rate 10 Mbit/s, 4B/5B, NRZI, and MLT-3 coding, two independent UTP or STP twisted pairs, category 5, star network topology using active Ethernet hubs on maximal segment length of 100m. The characteristic impedance of the twisted-pair cable is 100 Ω.

V. DISTRIBUTED CONTROL NODE

Based on the facts stated above, a distributed control node on the ARM STR912FAW44 has been implemented. The following description includes the current state of the development and it is necessary to point out, that the design work is still in progress and with regard to its complexity will continue in the future. The current solution described below is the application communication node featuring the two communication interfaces **CAN** and **Fast Ethernet**, equipped with the application layer specifications **MicroCANopen** and **HTTP web server**.

First in compliance with the requirements for the advanced distributed control node, the suitable hardware platform was chosen – ST Microelectronics ARM9 based 32-bit architecture microcontroller **STR912FAW44** meets the targets perfectly.

The STR912FAW44 main features from the point of the application are: ARM966E-S based 96 MHz core, 96 MIPS, 32-bit wide tightly coupled memory system with Harvard architecture - dual bank burst 512kB and 32 kB flash memory and 96 kB RAM memory, 5-stage pipeline pre-fetch queue and branch cache, JTAG interface boundary scan, ETM embedded trace module, configurable EMI external memory interface, 9 programmable DMA channels, VIC vectored interrupt controller - 32 sources, 8-channel 10-bit A/D converter, , 4x 16-bit Timers-Counters, Capture-Compare units, 3-phase induction motor controller – 3 pairs of PWM outputs and RTC real-time clock circuit. There are up to 80 configurable generally purposed inputs – outputs pins on the microcontroller.

From communication point of view the STR912FAW44 features 11 communication interfaces – two independent synchronous serial ports configurable as SPI, SSI and Microwire and, independent I^2C interfaces and USB 2.0 slave device interface. While these ones are assumed to be local board hardware interconnection or short office connection (USB), the next described interfaces are possible to use for long distance industrial communications. These are 3 UART channels (one of them with full modem control capability), CAN ISO 11898 2.0B interface, including 32 message objects, suitable for application protocol layer implementation and finally the MAC/DMA Ethernet controller with MII interface.

The software of the following application communication distribution control node was developed on the ST Microelectronics evaluation board STR910-EVAL containing all purposed appropriate communication interface circuits – the ISO 11898 high speed CAN transceiver and 100Base-TX Fast Ethernet transceiver. The structure of the developed application communication distributed control node on the STR912FAW44 is shown on the Fig.3.

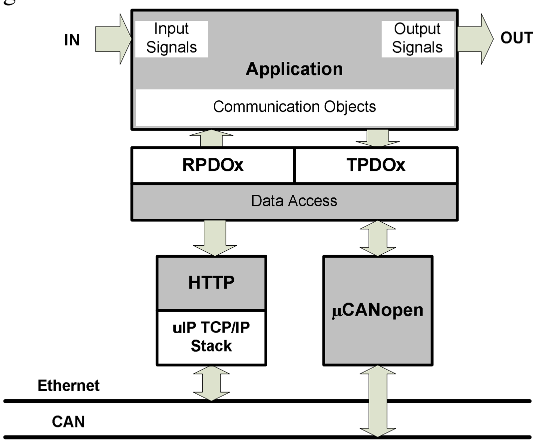

Fig.3. Distributed control node MicroCANopen – Ethernet HTTP

The main task of the distributed control node is the application algorithm processing. The **Application** reads all necessary digital and analog input signals **IN** (sensors, switches, buttons, status and warnings signals etc.) and updates output signals **OUT** (actuators, signaling and alarm elements, PWM outputs etc.) connected to the technological controlled system. The predefined input and output technological signal values, together with relevant internal status variables of the node are associated with the **Communication Objects**. These are directly mapped into the four receive process data objects **RPDO** (4 x 8 bytes) and the four transmit process data objects **TPDO** (4 x 8 bytes) layouts of the MicroCANopen protocol and exhibited to the **CAN** bus. These process data objects made up the application data interface of the MicroCANopen protocol and represent the part of the CANopen control network.

Through the second implemented communication interface **Ethernet** (Fast Ethernet optionally) it is possible at any time to view the RPDO and TPDO process data object of the MicroCANopen protocol stack. The monitoring of these ones is accomplished using implemented **HTTP web server** protocol above Ethernet and TCP/IP sub-layers. It is obvious that the HTTP web client requesting MicroCANopen protocol PDO objects of the distributed control node can be launched on any personal computer with any kind of web explorer running. HTTP 1.1 protocol version is supported. We can view the application example described above as the **HTTP monitor of the MicroCANopen protocol stack**. The Fig.4 shows the screen shot example of the client request result.

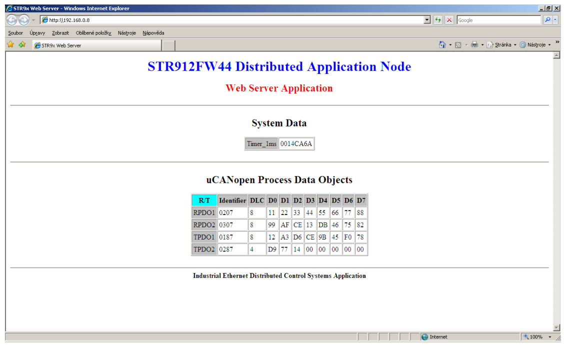

Fig.4. Internet Explorer client window example

It is desirable to mention that the example shown on the Fig.4 is possible to extend into any alternate option, in dependence on application requirements.

The application service program algorithm of the node is outlined on the Fig.5.

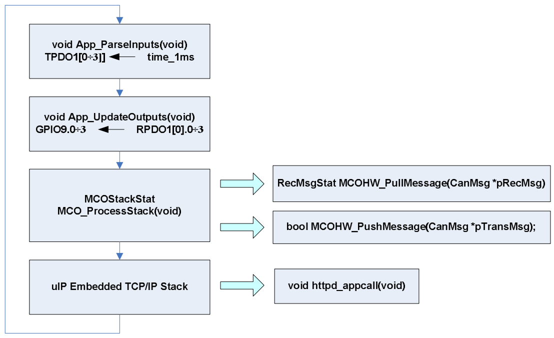

Fig.5. Application program cyclic service

It represents very simple approach to demonstrate the method of processing. The endless program loop serves step by step the following functional blocks:

App_ParseInputs, generally updating transmit process data objects by the technological inputs **IN** (Fig.3) or internal status variables values (internal 1 ms system timer is mapped into the TPDO1 in the example).

App_UpdateOutputs, generally updating technological outputs **OUT** (Fig.3) or internal status variables values by the receive process data objects (RPDO1 updates GPIO9 port of the STR912FAW microcontroller in the example).

MCO_ProcessStack, servicing the MicroCANopen process stack. It includes complete MicroCANopen stack implementation and interacts to the CAN interrupt service routine and the two following loopback functions, called if it is required. *MCOHW_PullMessage* provides transmit service request and *MCOHW_PushMessage* is called on CAN receive message event.

uIP Embedded TCP/IP stack serves complete Ethernet communication and implements the base TCP/IP protocol suite components, including IP, ARP, UDP and TCP protocol implementation. The example code calls the application loopback function *httpd_appcall*, in the case of IP (0x800) TCP (6) Ethernet frame type reception. The function implements on the default HTTP port (80) web server state machine. The data, making up the content of the web page (see Fig.4) and to be responded by the web server to client, is configurable and it is possible to change it.

VI. DATA NETWORK AND IDUSTRIAL BUS COMPARISON

Consistently with the goals of the paper, it is suitable to discuss the differences, identities, and relations between industrial communication bus and information data networks. A control network resembles in many ways a computer data network LAN - Local Area Network. The basics of the industrial communications and systems have been adopted from data networks concepts, where ISO/OSI data communication reference model completely describes structure and behaviour of a communication device. There are, however, some significant differences between data networks and industrial buses.

A **data network** is optimized for moving large amounts of data and the design of data network protocols assumes that occasional delays in data delivery and response are acceptable. The media access method used in data networks is mainly indeterminate. The complete protocol set of the ISO/OSI data communication reference model is implemented. Data networks operate usually in an office environment. In the most cases they are based on the Ethernet specification IEEE 802.3 - physical and link layer specification.
An **industrial bus** (often called fieldbus) on the other hand uses shorter data blocks to be transferred during short time periods in order to manage real-time responses of the system – the design of industrial control network protocols respects fast response requirements of control. The fieldbus media access method is mostly determinate. An industrial bus operates in a harsh environment, and increased electromagnetic immunity of the communication is required together with mechanical robustness. A fieldbus typically implements only physical, link, and application layers of the ISO/OSI data communication reference model.

VII. DATA AND INDUSTRIAL COMMUNICATION CONVERGENCE

In spite of the fact that the data network is not suitable well for the real-time processing and completely differs from the industrial communication bus, a lot of effort has been attended last time to utilize data networks for industrial communications and technological data transfer. The reasons are long distance internet network connectivity based on the standardized set of generally supported IP protocol specifications (TCP and UDP). Various gateways between an industrial bus and Ethernet network, industrial Web servers and terminal Ethernet devices are supplied by many vendors and companies. While a gateway between an industrial bus and Ethernet network enables connection between two different communication systems, industrial Web servers and terminal Ethernet devices view the Ethernet network as the fieldbus. This fact results in the increased requirements for implementation. In compliance with these requirements the conception of the **industrial Ethernet** has been defined. The necessary operating features are the increased electromagnetic immunity of the communication media, together with mechanical robustness, with regard to the harsh industrial environment. In dependency on the application, the real-time capability of the communication is required.

VIII. CONCLUSION

The expanding branch of industrial communications plays the key role in the future automation world and the proof typical industrial contemporary communication standards face the new progressive trends. There are communication solutions completely migrating to the new standards on one hand and, on the other hand, there are hybrid systems coexisting together.

The implementation project example of the application communication control distributed node was described. It demonstrated Ethernet communication exploitation for monitoring purposes of the industrial bus. Nevertheless it isn't definitive and it should result in additional projects, concerning direct using of Ethernet communication in industry, because the new advanced trend of the industrial communication systems (industrial Ethernet exploitation in automation) is more and more up-to-date. Due to its features the Ethernet specification is the future of industrial communications. It provides the long distance connectivity, the proof TCP/IP based protocol set, security, distant diagnostics and system reconfiguration. The remote long distance application algorithms debugging of distributed control systems is possible too, not speaking about smart adaptive distributed control systems. The chances of the Ethernet industrial exploitation are large and initiate extensive field of applications.

REFERENCES

[1] CAN Specification, Version 2.0. Robert Bosch, 1991

[2] CANopen Communication Profile For Industrial System Based on CAL, CiA Draft Standard 301 - Revision 3.0. CAN in Automation, 1996

[3] Pfeiffer, O., et. al.: Embedded Networking with CAN and CANopen. RTC Books, 2003. ISBN 0-929392-78-7

[4] Cisco Corp.: Internetworking Technologies Handbook – 1-58705-001-3

[5] IEEE 802.3 Standard – Part 3 – Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications