ACTA TECHNICA CORVINIENSIS – Bulletin of Engineering
Tome V (Year 2012). FASCICULE 3 [July–September]. ISSN 2067-3809

[1.] Peter PENIAK

CAN BASED APPLICATION PROTOCOLS FOR EMBEDDED DEVICES

[1.] UNIVERSITY OF ŽILINA, FACULTY OF ELECTRICAL ENGINEERING, DEPARTMENT OF CONTROL AND INFORMATION SYSTEMS, ŽILINA, SLOVAKIA

ABSTRACT: *Embedded systems are generally designed to perform the dedicated tasks with respect to device functions. Applications that are used in embedded systems are characterized by significant diversity with the different requirements for communication services. The interpretation of application data and control commands can be essentially different in interconnected embedded subsystems. The paper deals with CAN based application protocols that can be used for an interconnection of embedded devices via CAN fieldbus network. It is focused on selection of open application protocols that could be potentially used for device integration of different suppliers via CAN bus.*
KEYWORDS: *CAN, CAL, CIP, ZAL, ZCL, APS, NWM, API, ODVA, CiA*

INTRODUCTION

Embedded systems are generally designed to perform the dedicated tasks with respect to device functions. The small, computerized parts within a larger device then could serve for more general purpose and provide variety of functions within overall subsystem. These devices can be implemented either as standalone devices without necessity to interconnect with the other systems, or have to cooperate in real-time performance constraints.

Applications that are used in embedded systems are characterized by significant diversity with the different requirements for communication services. The interpretation of application data and control commands can be essentially different in interconnected embedded subsystems. Therefore using of the common open application protocols, which are independent on proprietary solutions of the particular vendors, is a key factor for a success. The particular manufacturers tend to support various open communication protocols, such as CAN, DeviceNet, Ethernet/IP and ZigBee. Usually, there are no problems when using devices of the same manufacturers for there is a good chance to use the same applications and their proprietary application protocols.

An example of CAN as an internal interface for interconnection of the embedded devices within printer subsystem is shown in Figure 1. The different subsystems are connected via CAN bus and can cooperate with using selected application protocol. Although CAN itself has not been used by all providers and we can still find a high technological heterogeneity in available products, due to its popularity and widespread deployment in automotive industry, we will focus on the possible open application protocols for CAN based embedded systems.

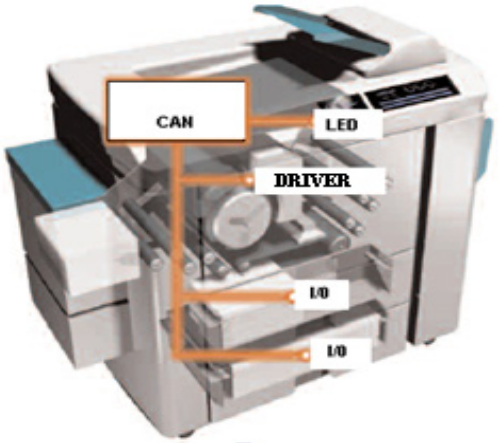

Figure 1. CAN internal interface for embedded printer subsystems

However, if there is a need to interconnect the devices of various manufacturers, the problems can occur despite using the same communication protocols. A vehicle control system can be used as a very good example as shown in Figure 2. CAN bus is very often implemented as integration layer for the devices of different vendors so that they can be interconnected with car's control system and on-board computer. Although a common application protocol has to be selected for their full integration.

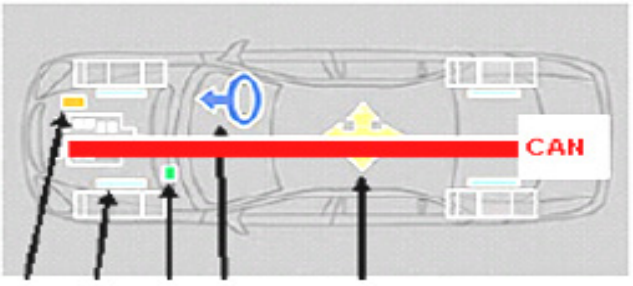

Figure 2. Vehicle control system with CAN

CAN APPLICATION PROTOCOLS

CAN application layer has to support implicit, explicit messages and provide mapping of CAN identifiers to the defined messages, or devices.

In addition, API (Application Programming Interface) is necessary for applications as well, providing the defined device profiles and necessary application objects. Device profiles and application objects represent virtual model of the defined devices and are used to identify capabilities of devices, map physical behavior to the variables and access application objects from applications.

Implicit and explicit application protocol messages with assigned CAN identifiers are encapsulated to the CAN frames and sent via CAN network. The network architecture and structure of the common application layer of CAN networks is shown in Table 1[1].

Table 1. CAN application layer

RM-OSI Layer	Network architecture (CAN based)	
API	Aplication objects Device profiles	
7	Implicit messages (I/O)	Explicit messages (I/O)
5, 6	(*Encapsulation*)	
4		
3		
2	CAN v.2	
1	ISO	

The application protocols used for embedded devices have not been unified yet. Individual vendors still try to use their proprietary architectures, however, there is an effort to create alliances of producers, such as ODVA, Fieldbus Foundation in order to promote a common standard for application layer [2,3]. The vendor independent standard CAL (CAN Application Layer) is supported by association CiA (CAN in Automation), while CIP (The Common Industrial Protocol) protocol for industrial automation applications is promoted by ODVA (Open DeviceNet Association).

As an alternative solution, ZigBee protocols ZAL (ZigBee Application Layer)/ ZCL (ZigBee Cluster Library) that are used in mobile applications and wireless networks, could be used as a potential solution for future as well. However, there have been no ZAL/ZCL implementations so far, except TCP/UDP encapsulation of ZAL/ZCL protocol messages [4]. Typical CAN/CAL application protocols that can be used with the CAN link layer services are shown in Table 2 [1]. ZigBee protocols are added as well, as a potential solution for integration with mobile devices.

Table 2. CAN based application protocols

RM-OSI Layer			
RM-OSI Layer	CAN/CAL protocols		
API	CAL	CIP	ZAL/ZCL
7			
5, 6	(*Encapsulation*)		
4			
3			
2	CAN v.2		
1	ISO		

CANOpen/CAL protocol

CAL provides a vendor independent application protocol for an object-oriented environment, which can be used for the integration of the various embedded devices. CAL protocol model is shown in Table 3.

Table 3. CAL application protocols

Layer	Protocol			
Device profiles	Profile A	Profile B	Profile C	……
7	CAL/CANopen (CiA)			
	NMT	DBT	LMT	CMS
-				
2	CAN 2.0			
1	CAN ISO 11898			

The protocol introduces a number of methods for transmitting and receiving messages through the so-called communication objects. CAL uses the following types of messages:

- *AM (Administration Messages)*
- *SDM/SDO(Service Data Messages/Objects)*
- *PDM / PDO (Process Data Messages / Objects)*
- *PM (Pre-defined Messages)*

PDO object is used for exchange of the implicit messages between applications. The objects are represented as variables, events (events), or data areas (domains), which are mapped to the corresponding PDO messages/objects in device directory. Always, the transmission is initiated by the client. SDO object is used for communication via explicit messages and supports communication in peer-to-peer mode. It can be used for non-fragmented messages with size 4B. Protocol is able to read and write data to object directory, transmit large volumes of data via fragmentation protocol.

Assignment of CAN identifiers is a key factor in the network architecture. CAN is able to control the priority of messages and provides a common list of identifiers (Poll), which are dynamically allocated to the end devices by object distributor (DMT). There is dedicated number of network identifiers (1260) available for objects (CMO) and only a small part is blocked for internal purpose.

CAL provides centralized management with network management server NMT, which basic task is the supervision of individual nodes in the network ("NMT Guarding"). NMT server maintains a list of active nodes and their status is periodically tested by message "Guard Request", as shown in Table 3. Device profiles are used for modeling of various network devices. CAL defines a common network directory for objects "Object Dictionary", in which every object is addressable by 16-bit index and 16-bit sub-index. The remaining CAL sub-protocols (DMT, CMS, LMT) will not be mentioned [1].

CIP PROTOCOL

CIP is object-oriented protocol for heterogeneous devices and networks (CAN, ControlNet, Ethernet/IP). The object model is applied for applications (API), but also in internal protocol architecture. It provides a wide range of communication services for embedded devices and supports the specific requirements of industrial and control applications [1].

CIP uses hierarchical model of communication and device addressing. Each device has assigned a unique identifier called MAC address (MAC ID #). The nodes are connected to the sub-networks interconnected to the CIP domain. CIP domain is seen as a logical area in which all nodes can be interconnected. To support specific communication requirements, the different domains can be established with assigned communication mode for each group independently (peer-to-peer, master-slave (cyclic), producer-consumer). There are two basic services:

- Explicit messages -for writing and reading of the object attributes, connection settings and file transfers
- Implicit Messaging – for exchange of /O process data in real time.

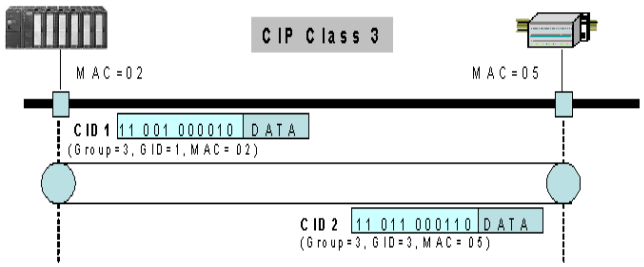

Figure 3. CIP addressing of CAL objects

The example of communication between two embedded devices via CAN and protocol CIP is illustrated by Figure 4. The format of CIP explicit messages is shown in Figure 10.

CIP Explicit Message

ID	DATA
11 b	8 B

0	GID		S –MAC ID	Group 1
1	0	MAC ID	GID	Group 2
1	1	GID	S-MAC ID	Group 3
1	1	1 1 1	GID	Group 4

GM	GID	MACID	F	XID	MAC ID	FT	FC	R/R	SC	A
2 b	3 b	6 b	1 b	3 b	6 b	2 b	6 b	1 b	n	

CIP header	
ID	*Identifier*
GM	*Group Message*
GID	*Group Message ID*
MAC ID	*MAC address*
S-MAC ID	*S MAC address*
F	*Fragment bit*
TID	*Transaction ID*
FT	*Fragment Type*
FC	*Fragment Count*
R/R	*Request/Response*
SC	*Service Code*
A	*Arguments*

Figure 4. Format of explicit CIP messages

ZigBee APPLICATION PROTOCOL (ZAL)

The ZAL protocol describes a set of structured communication primitives for an exchange of application data among mobile devices. ZAL protocol was designed for wireless communication of mobile devices via IEEE 802.15.4 link layer. The primary design goal of the ZAL was to support wireless communication of embedded systems, running on microcontrollers with limited amount of code, RAM and low network bandwidth. Therefore a compact low-traffic message format was developed with additional services, such as a device binding, service discovery and security protocol.

As a result, the ZigBee protocols became very popular and well-suited for a structured communication among networked embedded systems. There is a possibility that ZAL could be used as a common application protocol for CAN devices as well. Although the ZigBee Application Layer was originally designed to operate only over IEEE 802.15.4 wireless networks [6], an adaptation of the ZigBee Application Layer for CAN bus is generally possible. Figure 5 illustrates the possible use case. Mobile devices connected naturally use ZAL protocol for an internal communication via ZigBee. For a device integration, a dedicated gateway has to be used to link CAN devices with ZigBee ones. In addition, ZCL and APS (Application Support Sub-layer) application protocols are to be supported in CAN devices in order to enable full interoperability of the embedded devices.

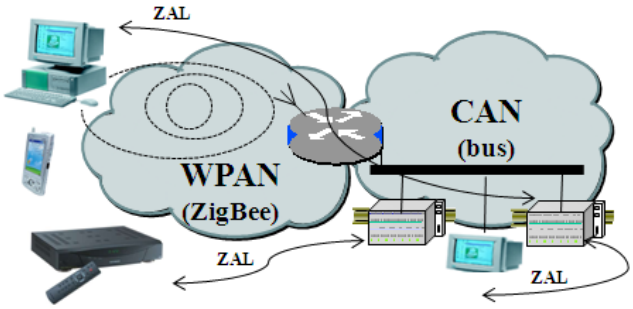

Figure 5. Integration of WPAN (ZigBee) and CAN devices

The integration approach has at least three major parts:

- ZigBee/CAN gateway for network integration
- Encapsulation of ZCL/APS messages via CAN bus protocol
- Mapping of addresses to CAN-ID identifiers

Figure 6 shows a model of ZigBee/CAN gateway. The gateway has physical interfaces for the both data link layers (CAN 2.0, IEEE 802.15.4).

In addition, the network layer protocol (NWM) is needed for ZigBee devices [5]. Finally, the main integration is to be covered by an additional application sub-layer (7+). Z-CAP protocol (ZigBee CAN Adaptation Protocol), responsible for address mapping and ZAL message encapsulation, is proposed to address the integration challenge.

The new protocol has to be implemented in ZigBee/CAN gateway and each CAN node (Fig. 6 b). However, ZigBee devices do not require any adaptation (Fig. 6 a).

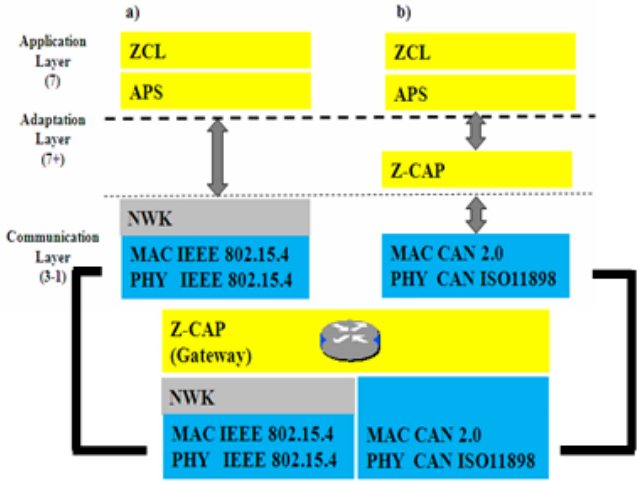

Figure 6. Protocol model of ZigBee/CAN integration with Z-CAP protocol

Z-CAP protocol offers two major services. Firstly, the mapping of ZigBee addresses to CAN-IDs is to be implemented. This process is based on Z-CAP binding table with assigned pairs of addresses (ZigBee, CAN). ZigBee devices would communicate with CAN devices as they would belong to ZigBee network. CAN applications would also use primarily ZigBee addresses. Z-CAP protocol just translates addresses to CAN-id and enables transfer of application messages among end nodes.

Secondly, an encapsulation of ZAL/ASP messages via CAN. Although an encapsulation is known technique, CAN propose very limited transport service with just 8B data left for a payload. It means, we cannot encapsulate the complete ZCL/APS messages (~70B) within one CAN frame (8B). Instead, ZAL message has to be divided to the several CAN frames and overall data integrity is to be managed by a fragmentation procedure. Z-CAP protocol therefore offers 1B of payload for fragmentation to identify all CAN frames belonging to 1 ZAL/ASP message (1B - FR=fragment id + FRC=fragment offset), as shown in Figure 7.

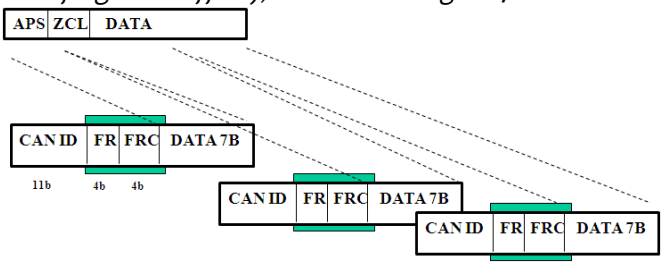

Figure 7. Encapsulation/Fragmentation of ZCL/APS messages via CAN

CONCLUSIONS

The paper dealt with CAN based application protocols that can be used for an interconnection of embedded devices via CAN fieldbus network. The main focus was on selection of open application protocols that could be used for integration of devices, with taking into consideration restriction to CAN link layer for transport of application protocol packets/messages. Besides the standard application protocols such as CAL and CIP, there is a new challenge with ZibBee application protocols that are much more popular and accepted by various vendors for mobile communication (WPAN). Therefore the recommendation is to focus on encapsulation of ZibBee protocols (ZAL, ZCS) via CAN messages in the same way as 6LoWPAN [RFC4944]. This would enable a simple integration with mobile embedded devices and could simplify overall network architecture of application layer for embedded devices.

REFERENCES

[1.] Franeková, M.- Kállay, F.- Peniak, P. Vestenický, P. (2007): Komunikačná bezpečnosť priemyselných sietí. ŽU Žilina, ISBN 978-80-8070-715-6 1.

[2.] Mahalik, N. P (2003).: Fieldbus technology, Industrial network standard for Real – Time Distributed Control, Springer, 2003

[3.] Kállay, F.- Peniak, P. (2003): Počítačové sítě LAN MAN WAN a jejich aplikace. Monografia, Grada Publishing 2003, ISBN 80-247-0545-1

[4.] Tolle G.(2008): A UDP/IP Adaptation of the ZigBee Application Protocol, October 2008, [WWW], http://tools.ietf.org/html/draft-tolle-cap-00

[5.] ZigBee specification, ZigBee alliance, January 17 2008, http://www.zigbee.org/Products/TechnicalDocuments Download/tabid/237/Default.aspx

[6.] IEEE 802 working group, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY), Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE Computer Society, Standard specification, http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf

ACTA TECHNICA CORVINIENSIS – BULLETIN of ENGINEERING

ISSN: 2067-3809 [CD-Rom, online]