BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Tomul LVII (LXI), Fasc. 3, 2011
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

DESIGN AND IMPLEMENTATION OF A CONTROLLER AREA NETWORK ON A HYBRID ELECTRIC VEHICLE EXPERIMENTAL MODEL

BY

DANIEL STICEA[*], GH. LIVINŢ and M. ALBU

“Gheorghe Asachi” Technical University of Iaşi,
Faculty of Electrical Engineering, Energetics
and Applied Informatics

Received, June 12, 2011
Accepted for publication: August 16, 2011

Abstract. This paper presents the design and implementation of a CAN network model of hybrid electric vehicle for distributed control of various systems founded on such a vehicle. CAN nodes are implemented with different numeric systems. Application-level communication was achieved through implementation of CANopen protocol.

Key words: CAN; hybrid electric vehicle; distributed control systems.

1. Introduction

Hybrid electric vehicles (HEV) seem to be a quite successful solution of a transitional phase before the advent of pure electric vehicles (green vehicles). These systems are far more complex than the ordinary vehicles and are composed mainly of specific elements borrowed from classic and electric automobiles. Thus, on a HEV there are many distributed control subsystems subordinated to a supervisory system.

[*] Corresponding author: *e-mail*: danisticea@yahoo.com

Because many systems need to be interconnected on a vehicle, there may be more communication networks depending on the role and importance of these systems. In order to assure the control of the propulsion and its subsystems, a communication network drive, *i.e.* Controller Area Network (CAN) of high speed (1 Mb/s), is generally used. The following subsystems fall into this category: engine heat control, the electric motor, the transmission, the energy management system, the diagnostic system, as well as the supervisory and control systems.

2. CAN Communication

CAN is a serial communication protocol used for the distributed and real time control systems. It first appeared in 1983 and was designed by Bosch for the automotive industry, being implemented at a low cost and able to cancel the electromagnetic disturbances usually founded in the vehicle. It allows more than one master node and can operate at speeds of up to 1 Mb/s (Richards, 2000). It consists of a network on several levels similar to an OSI model connection (Open Systems Interconnection). The physical layer is covered by the CAN standard ISO11898-2, known as *high-speed CAN* (Cook & Freudenberg, 2007).

A protocol known for the application layer of the CAN network is the CANopen protocol. It uses for its implementation the message identifiers and the main data package from the message structure. The allocation of a CAN node ID is taken from the CAL protocol (CAN Application Layer). However, the CANopen brings as novelty the dictionary of objects, what is an ordered data structure. This structure is used as a way of transferring the information between the CAN communication network and the application.

3. The Hybrid Electric Vehicle Experimental Model

The parallel hybrid electric vehicle experimental model configuration is shown in Fig. 1. It includes an engine F8Q of 1.9 L capacity, 64 HP and an electric traction motor. The engine is connected through an automated manual gear boxes to a torque distributor. The electric machine is a squirrel cage asynchronous machine (15 kW, 380 V, 30.5, 50 Hz, 2,940 rpm) supplied by the PWM inverter. The emulation of the load relating to the hybrid electric vehicle model was accomplished with an induction motor controlled in couple by a Siemens back-to-back converter (Sinamics S120) connected to a three-phase power network.

The energy storage system is made out of 24 batteries with Pb (12 V/45 A) interfaced with the DC network on the vehicle through a bidirectional DC / DC converter. In order to manage the battery system, a dsPIC30F4013 numerical system was used.

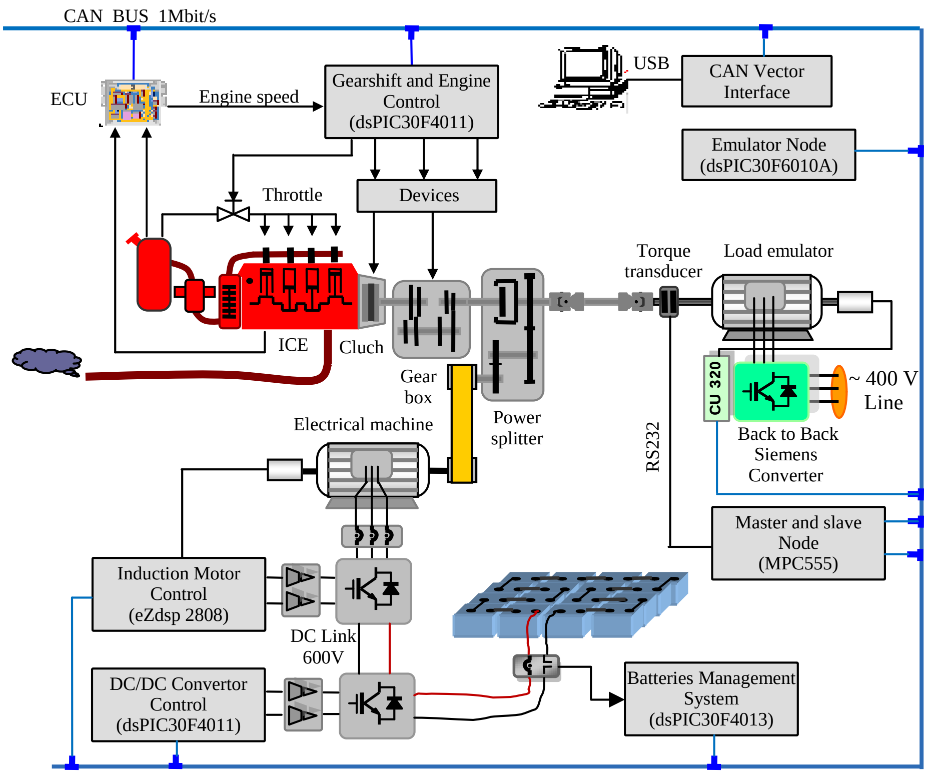
CAN BUS 1Mbit/s
ECU
Engine speed
Gearshift and Engine Control (dsPIC30F4011)
USB
CAN Vector Interface
Emulator Node (dsPIC30F6010A)
Throttle
Devices
Torque transducer
Load emulator
ICE
Cluch
Gear box
Power splitter
CU 320
~ 400 V Line
Back to Back Siemens Converter
RS232
Electrical machine
Master and slave Node (MPC555)
Induction Motor Control (eZdsp 2808)
DC Link 600V
DC/DC Convertor Control (dsPIC30F4011)
Batteries Management System (dsPIC30F4013)

Fig. 1 – The hybrid electric vehicle experimental model.

4. CAN Network Distribution on HEV Experimental Model

The distribution of the CAN network at the level of the experimental model is shown in Fig. 2. The COB-ID (Communication Objects Identifier) allocation model for each node was statically made by assigning an ID to each node in the network design phase.

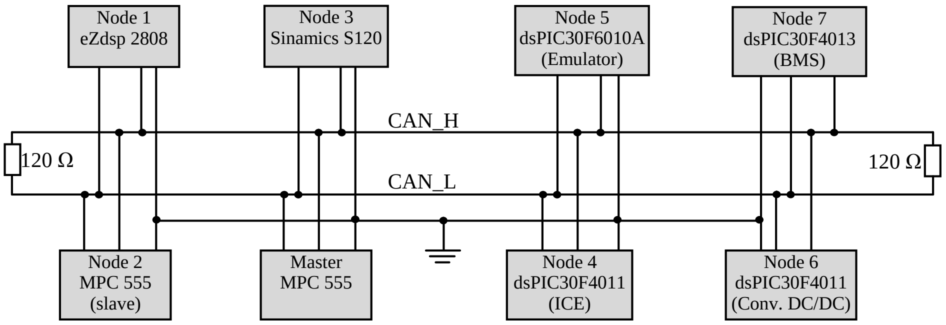
Node 1 eZdsp 2808
Node 3 Sinamics S120
Node 5 dsPIC30F6010A (Emulator)
Node 7 dsPIC30F4013 (BMS)
CAN_H
120 Ω
CAN_L
120 Ω
Node 2 MPC 555 (slave)
Master MPC 555
Node 4 dsPIC30F4011 (ICE)
Node 6 dsPIC30F4011 (Conv. DC/DC)

Fig. 2 – The assignment mode of the CAN nodes on the VEH stand.

Node 1 was assigned to the numerical system eZdsp used to the electric machine control of the vehicle.

Node 2 was assigned to the DTR torque transducer that measures the torque of electric motor or engine in the range 0…250 N.m. The information is sent on the RS232 interface to the MPC555 numeric system where it is received and interpreted and then forwarded to the CAN network.

Node 3 was assigned to the control system of the asynchronous electric motor in order to emulate the vehicle’s dynamics.

Node 4 was assigned to the numerical system for internal combustion engine control (ICE). Besides the control of the engine’s torque, this node has also the mission to control the automatized change of speed. This implies the control of the engine’s acceleration, of the clutch and of the effective change of the speed. The numerical system used in order to implement this node is a dsPIC30F4011 micro-controller with resources and peripheries appropriate to engine control and CAN communication interface.

Node 5 was assigned to the numerical system implemented on dsPIC30F6010A microcontroller that conveys the torque for the asynchronous electric motor that emulates the real dynamics of the vehicle. This reference is calculated online based on the already measured speed, on the vehicle’s parameters and on the imposed road conditions.

Node 6 was assigned to the numerical system in order to assure the control of the DC-DC converter; this one is used on the interface between the batteries and the CC bus (600 V) of the stand.

At last, node 7 was assigned to the management system for the battery. The master node was implemented on the MPC555 numerical system and has the role to supervise the entire activity of the local control systems.

5. CANopen Implementation Protocole on the Experimental Stand

Implementing the CAN protocol has been made using the graphic programming (Matlab/Simulink blocks) combined with the textual programming in C++. For example, in Fig. 3 it is represented the implementation of the node 4 inside the network using the Matlab/Simulink method.

The first line of the Fig. 3 contains the initializing blocks that belong to the peripheries used by the controller for the adjustment of the engine’s torque and of the automated speed changes (PWM Config, ADC Config, Timer Config and Port Config). These blocks are dialed just one time at the beginning of the program, as well as the initializing blocks (InitCANOpen) and configuration blocks (CAN configuration) of the CAN node. Then, at every milisecond generated by the SW_TimerISR timer, the CAN stack is dialed using the From/To CANOpen Stack block. This ones verifies whether new messages have been received or whether new messages are prepared in order to be sent (CANOpen Message Send). After that, the cCall function is dialed in order to implement the actual control. The CANOpen Err&Run LED’s is used to signal the CAN node’s status by the illumination of two light emitting diodes.

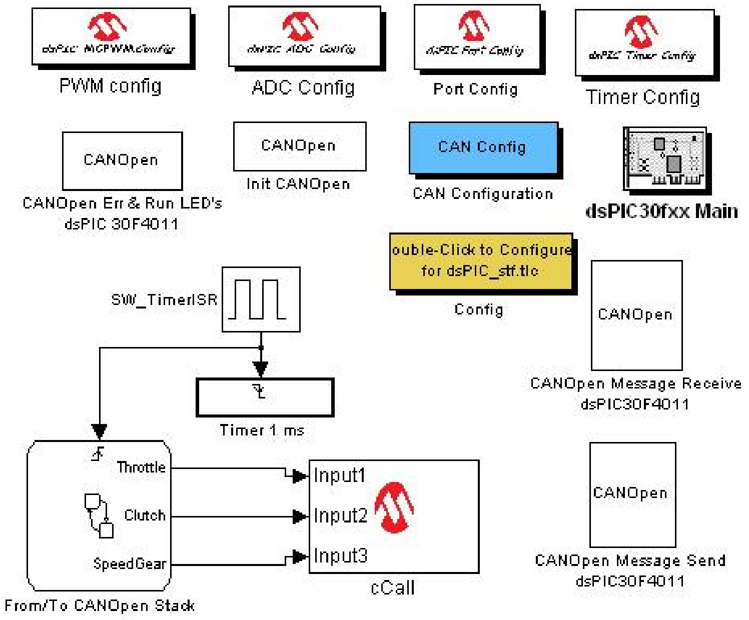

Fig. 3 – The implementation mode of node 4 on the VEH stand.

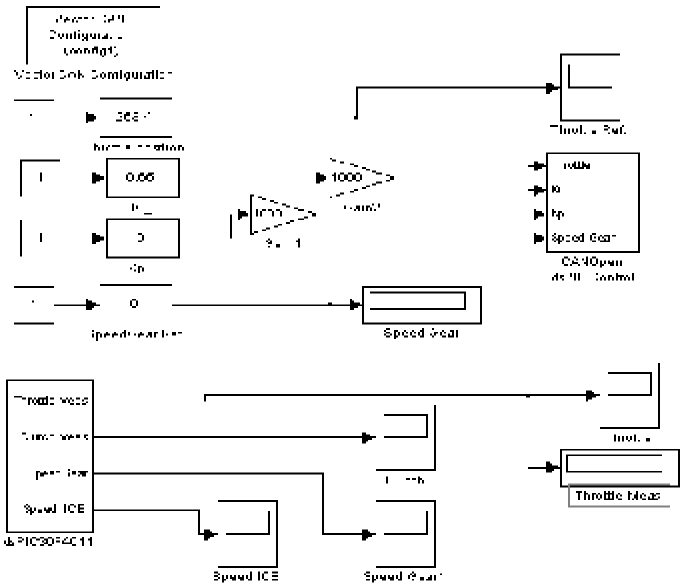

Fig. 4 – Node 4 interface of the CAN network with the PC.

Fig. 4 presents the PC graphical interface with node 4 of the CAN network. The connection of the PC to the CAN communication network from the experimental stand has beed established using the CANVector specialized device. The interface program has been also developped in the Matlab & Simulink environment. This one permits the PC's intervention on the CAN and the storage by a certain node of messages sent by the CAN network. In Fig. 5 the engine response to a step signal of the reference can be observed. The engine is controlled in torque by a PI regulator with saturation and anti-windup.

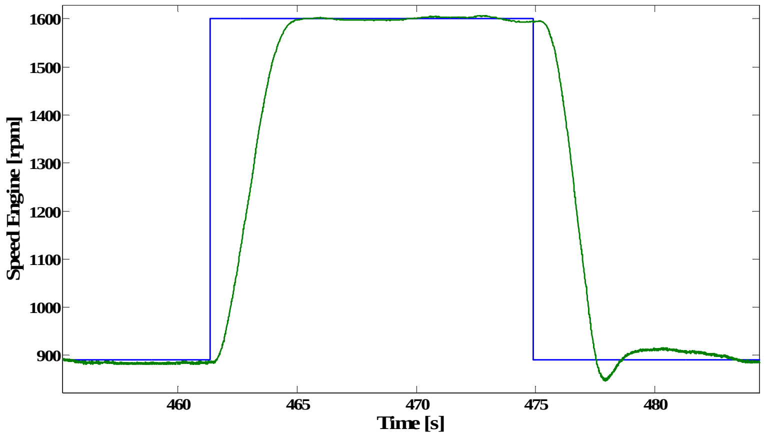

Fig. 5 – The engine's response at step signals.

6. Conclusions

CAN is a network that accommodates itself very well in a hostile environment inside the vehicles and can fulfill all the demanding conditions for a distributed control in real time.

Loading the network depends on the number of nodes from the network and on the number of data packs transmitted at very short time intervals. A node can succesfully transmit data packs by the milisecond without having any difficulties. The CAN network described and implemented on HEV model answers all this requirements.

REFERENCES

Boterenbrood H., *CANopen High-Level Protocol for CAN-Bus.* NIKHEF, Amsterdam, 20 March, 2000.

Cook J.A., Freudenberg J.S., *Controller Area Network (CAN).* EECS 461, Fall 2007.

Lian J.**,** Zhou Y.**,** Sun X.**,** Tian B., Lin C., *Design and Implementation of Hybrid Electric Vehicle Control System CAN Communication.* Adv. Mater. Res., **179-180**, 359-364 (2011).

Livinţ Gh., Horga V., Răţoi M., Albu M., Chiriac G., *Implementing the CANopen Protocol for the Distributed Control of a Hybrid Electric Vehicle*. Proc. of 8th Internat. Symp. on Adv. Electromech. Motion Syst., Lille, July 1-3, 2009.

Richards P., *A CAN Physical Layer Discussion*, Microchip Technology Inc, Application Note, AN228, 2002.

PROIECTAREA SI IMPLEMENTAREA UNEI RETELE CAN PENTRU CONTROLUL DISTIBUIT AL UNUI MODEL EXPERIMENTAL DE VEHICUL ELECTRIC HIBRID

(Rezumat)

Se prezintă elemente de proiectare şi implementare a reţelei CAN pe un model experimental de vehicul electric hibrid, pentru controlul distribuit al diferitelor sisteme întâlnite pe un astfel de vehicul. Nodurile CAN sunt implementate cu diferite sisteme numerice. Comunicarea la nivel de aplicaţie a fost realizată prin implementrea protocolului CANopen.