AiMT Advances in Military Technology
Vol. 11, No. 1, June 2016

Application for Designing Communication Algorithm Using CAN with CANaerospace Protocol

P. Janů*

Department of Radar Technology, University of Defence, Brno, Czech Republic

The manuscript was received on 12 February 2016 and was accepted after revision for publication on 8 June 2016.

Abstract:

This paper introduces the most common data buses used in on-board aircraft electronic systems focusing on affordable buses and the reason for their use. Controller Area Network (CAN) was selected as the main bus representative. CAN has its special use in avionic systems called CANaerospace. In avionic systems, it is necessary to access the bus at precisely defined time intervals according to the schedule called Cycle Matrix. There is a need to develop an avionic system application for designing Cycle Matrix. Therefore, this paper focuses on mathematical analysis and a detailed description of the user application for designing Cycle Matrix in MATLAB environment. The application is intended for CAN bus specialists and it is supposed to assist with the design of communication algorithm for any avionic system.

Keywords:

Bus utilization, CAN, CANaerospace, Communication analysis, Cycle matrix, MATLAB

1. Introduction

Currently, military combat and transport aircraft, as well as civilian aircraft use very specific data buses to acquire required parameters from various sensors all over the aircraft. The main bus representatives are ARINC429, ARINC629, AFDX, MIL-STD-1553, MIL-STD-1773 and IEEE 1394b [1]. The main disadvantage of these buses is their cost. Each module of the aircraft electronic system needs a device that allows the module to connect to the specific bus and to ensure the communication via this bus. For this reason, there is a general effort to implement a data bus to the aircraft electronic systems that would lower this economic burden. Controller Area Network (CAN) [2] is one possible candidate for avionic data acquisition system.

* *Corresponding author: Department of Radar Technology, University of Defence, Kounicova 65, 662 10 Brno, Czech Republic. Phone: +420 973 4451 94, E-mail: premysl.janu@unob.cz*

CAN offers numerous advantages, such as that most microcontrollers (that can be used in on-board aircraft electronic systems) have a device for establishing communication already implemented. Another very important advantage is the communication method itself. The communication is not based on addressing individual modules, as it is in case of the above mentioned buses. CAN has an identification number assigned to each message. The message is then sent to the bus and received by all other nodes. Each node of the system has a pre-set filter that decides whether the given message will be processed or not. Therefore, it is possible that more nodes receive one message and work with it at the same time. CAN also allows two approaches of communication, so-called event-triggered and time-triggered. For the event-triggered mode, the message is received by the bus if the bus is free at that moment. If two messages are trying to enter the bus at the same time, the one with a lower identification number, thus higher priority, will be used. In the time-triggered mode, a precise timetable for messages entering the bus is determined.

Each aircraft electronic system provides a huge amount of data that is split into several phases. It is strongly recommended for the used modules, units or sensors of the complete system to transmit their data only at an exact time. A time-triggered communication mode is the best mode for this purpose. The communication is running according to a pre-set schedule, called Cycle Matrix. Cycle Matrix is shown in Fig. 1.

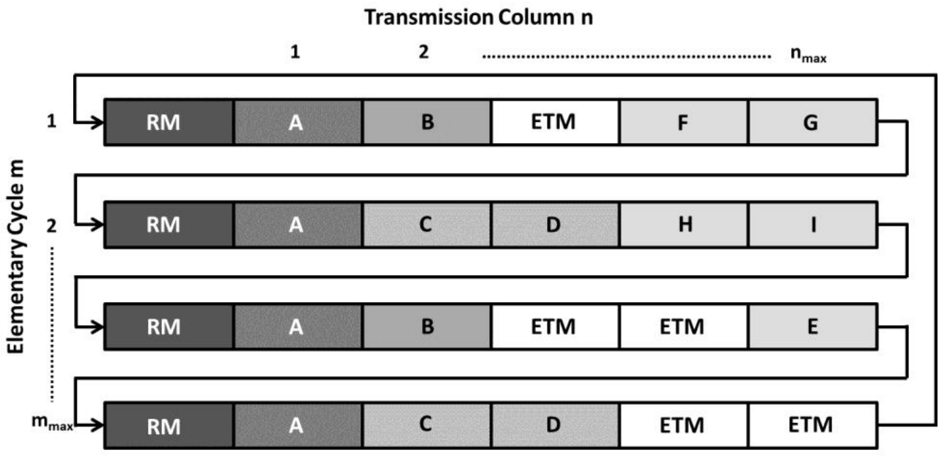


Fig. 1 Cycle Matrix

The Cycle Matrix consists of a number of rows and columns. Each row of the Cycle Matrix is called an Elementary Cycle. The Elementary Cycle is always pre-phased by a Reference Message (RM). The RM has a high priority and ensures the correct synchronization of the communication via the bus. Individual messages (that have a specific location/ a time slot in the defined communication algorithm and are sent periodically) follow after the synchronization message. Some empty time slots for eventual broadcast during the event-triggered mode (ETM) also appear in the Elementary Cycle. This communication algorithm is continuously repeated for one flight phase and it is different for each flight phase.

CANaerospace protocol [3] was specified for the use in avionic systems. It defines the form of the messages and data formats for highly reliable communication of on-board aircraft systems. CANaerospace protocol has been used in several projects

(Ae270, SATS, V220 and V300T aircraft engines and SOFIA). The latest project, in which the CANaerospace protocol was used, was SimStar Bravo flight simulator [4]. It was developed by the Faculty of Information Technology in Brno University of Technology, in cooperation with Evektor Company and VR Group Company in 2014. It is a light sport aircraft simulator.

The CANaerospace protocol specifies five types of messages, whose importance and purpose is divided by the CAN identifiers. Each type of message is assigned to a certain data type and each type of message has a characteristic structure.

2. Mathematical Principle

In order to create an application for designing and analyzing the communication algorithm utilizing CAN and CANaerospace protocol, it is important to know the mathematical principle. In addition, the bus capacity and the bus utilization are very important parameters. The CAN bus allows working with 11 or 29-bit identifier, which also indicates the number of possible transmitted messages. There are 2048 messages for 11-bit identifier, which is an adequate number according to CANaerospace specification. Thus, the mathematical apparatus will be based on the 11-bit identifier. For the following calculations, it is also required to know the length of the data frame n_D. That is given by the following equation:

$$n_D = round\left[\frac{34+8N_D}{5}\right]+47+8N_D\,, \tag{1}$$

where N_D is the amount of data bytes.

Function `'round'` rounds down to the nearest integer. It is because of the synchronization mechanism, so-called insertion of stuff bits.

Transmission time of one bit τ is defined as a reciprocal value of the baud rate:

$$\tau = \frac{1}{baudrate}\,. \tag{2}$$

Another very important parameter for the bus capacity calculation is the message transmission period. It is measured at 100% bus utilization, thus, when messages are transmitted immediately one after another. The period of a message transmission $p_{U100\%}$ is defined as:

$$p_{U_{100\%}} = n_D\tau\,. \tag{3}$$

Bus capacity C_{CANAS} represents the amount of messages that can be sent over the bus per second. This is defined by the following equation:

$$C_{CANAS} = \frac{1}{p_{U_{100\%}}}\,. \tag{4}$$

The bus capacity depends on the number of complementary bits inserted during the process of bit stuffing. Amount of the stuff bits is very stochastic.

The last very important parameter which indicates the current status of the bus is the bus utilization. There are two types of messages that can be transmitted over the bus with a different formula for calculation of each type of message. The first type is a message that is sent periodically to the bus and it is called a synchronous message. This message can be assigned specifically to the time-triggered mode. The second type is a message that is transmitted randomly to the bus and is called an asynchronous message. This message can be assigned to the event-triggered mode. The bus utilization U_S for synchronous messages is defined as:

$$U_S = n_S \tau \sum_{i=1}^{N} \frac{1}{p_{S_i}} 100 , \qquad (5)$$

where n_S is the data frame length of the synchronous transmitted messages, τ is the transmission time of one bit, p_{si} is the synchronous message transmission period and N is the amount of transmitted messages.

Bus utilization U_{AS} by asynchronous messages is defined as:

$$U_{AS} = n_{AS} \tau \sum_{i=1}^{N} x_{AS_i} 100 , \qquad (6)$$

where n_{AS} is the data frame length of asynchronous transmitted messages, τ is the transmission time of one bit, x_{AS} is the amount, how many times was a certain asynchronous message transmitted and N is the amount of asynchronously transmitted messages.

CANaerospace specification recommends 80% bus utilization by the synchronous messages to leave some bus capacity for transmitting asynchronous messages, especially for those that are used for reliable bus communication control and emergency messages.

3. User Application for Designing Communication Algorithms Utilising CAN Bus with CANaerospace Protocol

MATLAB development environment was chosen as the user application for both designing communication algorithm and for analysis utilizing CAN bus and CANaerospace protocol. There are two ways of creating user applications in MATLAB. The first is using a GUIDE and the second one is using a method of switched board programming [5, 6]. The switched board programming method was used for this user application. The created application was aimed at a user, who understands CAN, communication methods via bus and CANaerospace protocol.

The main application window shown in Fig. 2 contains individual edit frames for important data needed for designing communication algorithm; frames for calculated data display; and pushbuttons necessary for application control. First, the user gradually enters periods of transmitted messages and their values are entered in milliseconds. They are synchronous messages. The CANaerospace specification contains 736 messages that can be periodically transmitted. At a baud rate of 125 kbit/s - if all the messages were transmitted with the same transmission period - the period of one message would be 92 ms. At a baud rate of 500 kbit/s, the period for one message would be 23 ms. If the baud rate was 1 Mbit/s, which is the maximum that the CAN bus allows, the period for one message would come out at a value of 11.5 ms. All the 736 messages cannot be transmitted simultaneously, which is related to the aircraft flight phases, as some variables would be irrelevant for a particular flight phase. Therefore, the minimum period for one message was defined as 12.5 ms. This is a sufficiently low value and the user enters multiples of this value for a simple design of the Cycle Matrix. To give an idea - with this transmission period and with 80 % bus utilization at a baud rate of 125 kbit/s, it is possible to transmit 10 messages; 40 messages at a baud rate of 500 kbit/s. The CAN bus theoretically allows a baud rate of 1 Mbit/s, but at such a high baud rate connection problems have arisen. This fact was tested in [7]. After entering the message transmission periods, the user confirms the data with a pushbutton. The assigned period is written to a vector and the value is cleared from the edit window. The entered transmission periods are displayed next to each other underneath the edit window to clearly show what message transmission periods were entered. Amounts of each message period are displayed on the side of the edit window.

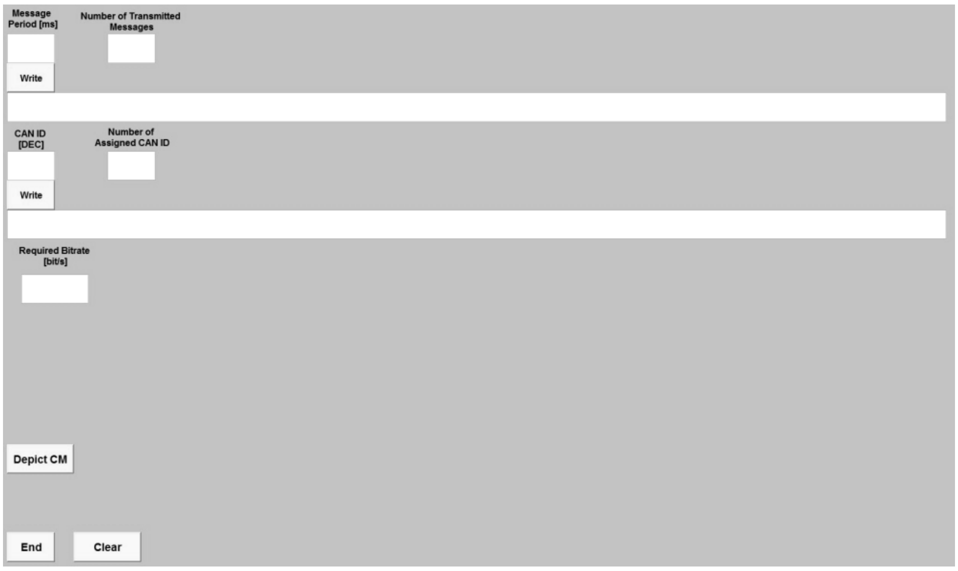


Fig. 2 Main application window

Then, the identifiers for each message must be determined and entered. The selection of each identifier is confirmed by a pushbutton, then inserted into the vector and its value is deleted from the edit window. Values are entered in a decimal integer and all entered identifiers are displayed next to each other under the confirmation button. Again, the amount of each entered identifier is displayed. This amount must match the amount of entered message periods. A corresponding baud rate is displayed in the window below the line of identifiers. This baud rate is calculated and displayed simultaneously with each submission of the transmission message periods. This enables a simultaneous analysis of the potential number of transmission messages. The required baud rate *bitrate* is calculated according to the relation:

$$bitrate = \frac{n\sum_{i=1}^{N} f_{p_i} 100}{80}, \tag{7}$$

where n is the amount of bits in the message, N is the amount of messages transmitted, f_{pi} is the transmission frequency of one message.

Furthermore, the main application window contains three important pushbuttons. One button resets all entered and calculated values before the next entry. The exit button closes the entire application. The third and most important button is for displaying the Cycle Matrix. This button controls the entire process of designing the communication algorithm.

After the button is pushed, the minimum entered message period, stored in the vector, is selected. This period corresponds to the period of synchronization message and is added to the saved vector of the message periods. Firstly, the amount of sent messages with the synchronization message is shown. Secondly, after adding a reference message, the required baud rate is calculated and displayed. Thirdly, the program determines a realistic baud rate that the microcontrollers can work with. From these, a vector of baud rates is defined, which can be implemented by microcontroller *CANBR = [100 000 125 000 200 000 250 000 500 000 1 000 000]*. Values are in bit/s. By using multiple branching, the needed baud rate is assigned to the next higher value from the vector *CANBR*, i.e. the real value of the baud rate. From this, the duration of one time slot *DTS* is calculated in ms. *DTS* is another important element for the Cycle Matrix design. It is calculated according to the following equation:

$$DTS = \frac{n}{bitrateR}1000\,, \tag{8}$$

where n is the amount of bits in the message and *bitrateR* is the realistic baud rate applicable to the microcontroller.

The value *DTS* is displayed too. Then, the amount of time slots per one period of the transmitted message *PTS* is calculated in ms. This is needed for the future Cycle Matrix dimension derivation and is calculated as follows:

$$PTS = round\left(\frac{1000}{f_{p_i} DTS}\right), \tag{9}$$

where f_{pi} is the frequency of the given message and *DTS* is the duration of one time slot. Function `'round'` rounds the number *PTS* to the nearest lower integer.

Subsequently, recalculation of the message periods is performed according to one time slot. For designing the Cycle Matrix, it is important to determine the Cycle Matrix size. The amount of columns is equal to the minimum amount of time slots per one message period `min(PTS)`. The number of rows is derived by rounding down the following ratio `'round(max(PTS)/min(PTS))'`, (i.e. a ratio of the maximum amount of time slot to the minimum amount of time slots) to the nearest lower integer. This number corresponds to the longest period of the transmission message. The longest message period thus indicates the number of elements in the Cycle Matrix. Computation of the bus utilization U is calculated according to the following equation:

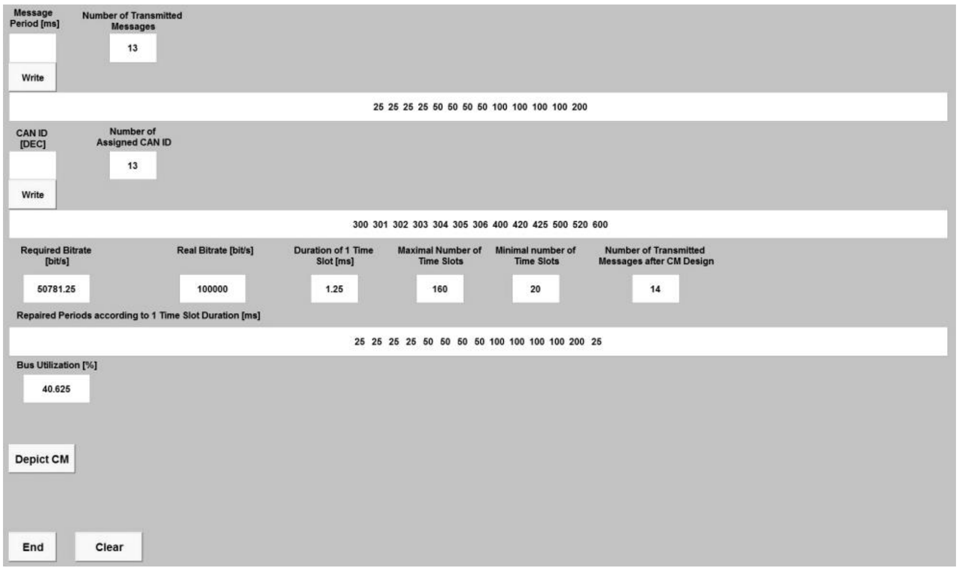
$$U = \frac{n}{bitrateR}\sum_{i=1}^{N} f_{p_i} 100\,, \tag{10}$$

where n is the amount of bits in the message, *bitrateR* is the realistic baud rate applicable to the microcontroller, N is the amount of transmitted messages and f_{pi} is the transmission frequency of a particular message.

All the important parameters, such as the re-calculated message periods, the minimum and maximum amounts of time slots and the total bus utilization are also displayed. For identification of what position each individual message will have in the Cycle Matrix, multiple numbers of the minimum period are calculated and composed into a vector.

A zero matrix is created as a first Cycle Matrix draft. Subsequently, the first column is replaced by number 129, which is the identifier value for a reference message. Then, a vector is created that contains coordinates of zeros inside the Cycle

Matrix. After that the Cycle Matrix is gradually calculated by indexing according to this vector and the vector of the minimum period multiples. The minimum period multiples actually determine the row of the Cycle Matrix. Individual identifiers of the periodic messages are placed at positions where zeros were previously placed until the identifiers run out. The positions where zeros are left are filled by an ETM string (Event Triggered Mode) and the number 129 is substituted by an REF string (REFerence message). Finally, the proposed Cycle Matrix is rendered. The completed application window with entered and computed values is shown in Fig. 3.



Number of Transmitted Messages: 13

25 25 25 25 50 50 50 50 100 100 100 100 200

Number of Assigned CAN ID: 13

300 301 302 303 304 305 306 400 420 425 500 520 600

Required Bitrate [bit/s]	Real Bitrate [bit/s]	Duration of 1 Time Slot [ms]	Maximal Number of Time Slots	Minimal number of Time Slots	Number of Transmitted Messages after CM Design
50781.25	100000	1.25	160	20	14

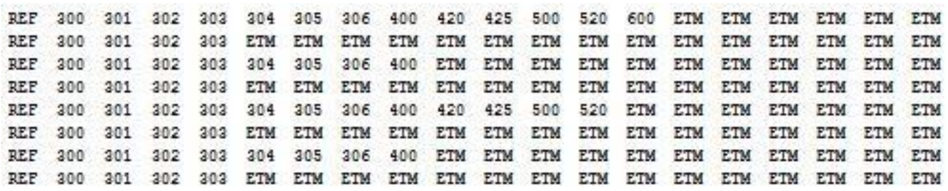
Repaired Periods according to 1 Time Slot Duration [ms]

25 25 25 25 50 50 50 50 100 100 100 100 200 25

Bus Utilization [%]: 40.625

Fig. 3 Application window with entered and computed values

The outcome of the above created user application is the designed Cycle Matrix. Example of Cycle Matrix form is shown in Fig. 4.



REF	300	301	302	303	304	305	306	400	420	425	500	520	600	ETM	ETM	ETM	ETM	ETM	ETM
REF	300	301	302	303	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM
REF	300	301	302	303	304	305	306	400	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM
REF	300	301	302	303	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM
REF	300	301	302	303	304	305	306	400	420	425	500	520	ETM	ETM	ETM	ETM	ETM	ETM	ETM
REF	300	301	302	303	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM
REF	300	301	302	303	304	305	306	400	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM
REF	300	301	302	303	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM	ETM

Fig. 4 Designed Cycle Matrix created by the new user application

4. Conclusion

This paper described data buses used for acquisition of variables from aircraft sensors during different flight phases, as well as it introduced and described a new trend of using economically less demanding data buses such as CAN bus with CANaerospace protocol. In addition, two methods of communication were presented. One of them is particularly important for on-board aircraft electronic systems, and it is called the time-triggered communication method. Mathematical principles of this communication method were also explained. A user application for designing communication algorithm, called Cycle Matrix, was presented as the core idea of this publication. The main purpose of the above described application is designing and displaying the final Cycle Matrix, but the application allows much more. It also enables an analysis of the communication via the bus. It can crosscheck the needed baud rate using the entered message periods and evaluate how many messages can be entered to meet the CANaerospace protocol requirements. It can determine what realistic baud rate should be set in the microcontroller. It can track the amount of messages that will be transmitted, the communication algorithm dimension and the total utilization of the bus for the user-proposed Cycle Matrix. Therefore, the above described application significantly helps users to design communication algorithms for on-board aircraft electronic systems. There are no known methods for designing Cycle Matrix using any available software. So far, the designer had to solve all the important issues connected with the Cycle Matrix design on a sheet of paper. It included many calculations using the above mentioned equations. Therefore, this new application saves a lot of time and it also provides analysis/feedback of the bus communication. In the future, a worthwhile improvement to the application is foreseen, as to make a database of all the flight and aircraft parameters according to the identification numbers of CANaerospace protocol which would make the application even more user friendly. The application would then display a list of the transmitted variables in the Cycle Matrix according to its identifiers.

Acknowledgement

This paper was created with a support of the project PROKVES of University of Defence, Brno - Research of Sensory and Control Systems for Information Dominance on a Battlefield.

References

[1] JANU, P. Analysis of CANaerospace Protocol Communication Quality in Aviation System. *Advances in Electrical and Computer Engineering*. Romania: Romanian Academy of Technical Sciences Stefan cel Mare University of Suceava, 2014, vol. 14. no. 1, p. 81-86. DOI: 10.4316/AECE.2014.01013. ISSN 1582-7445.

[2] VOSS, W. *A Comprehensible Guide to Controller Area Network*. 2nd. ed. Greenfield, Massachusetts, USA: Copperhill Technologies Corporation, 2005-2008. p. 152. ISBN 978-0-9765-1160-1.

[3] CANaerospace - the Airborne CAN Interface Standard. *Stock Flight Systems* [online]. Germany: Stock Flight Systems, 1993 [cit. 2016-01-19]. Available from: http://www.stockflightsystems.com/canaerospace.html.

[4] Flight Simulator SimStar Bravo Presented by Evektor at the Exposition Aero Friedrichshafen. *Evektor* [online]. Kunovice: Evektor, 1992 [cit. 2016-01-19]. Available from: http://www.evektor.cz/cz/novinky/letecky-simulator-simstar-bravo-prezentovan-evektorem-na-vystave-aero-friedrichshafen#.Vp4BQk_nUik (in Czech).

[5] ZAPLATILEK, K. and DONAR, B. *MATLAB for Beginners*. 2nd. ed. Prague: BEN – Technical Literature, 2009. p. 152. ISBN 80-7300-175-6 (in Czech).

[6] ZAPLATILEK, K. and DONAR, B. *MATLAB Creation of User Applications*. 1st. ed. Prague: BEN – Technical Literature, 2008. p. 216. ISBN 978-80-7300-133-9 (in Czech).

[7] JANU, P. *A System of Data Acquisition and Processing from Aircraft Onboard Systems*. Brno, 2011. Dissertation. University of Defence. Supervisor prof. Ing. Rudolf Jalovecky, CSc.