CAN/TTCAN simulator system based on educational DSP hardware kits

A.Hara, K.Fonseca, L. Scandelari
CPGEI/CEFET-PR
Av.Sete de Setembro 3165 Curitiba-PR, BRAZIL
+5541310-4695 (phone) 4683 (fax)
{hara,keiko}@cpgei.cefetpr.Br, Luciano@cefetpr.br

Simulation is useful on designing industrial communication systems and also a powerful teaching support tool. This work presents a CAN simulator system built on educational DSP hardware kits. Its design allows the study of network temporal behavior and its performance under a message workload. This is done by checking on the bus hardware several variable values for different bus parameter configurations. A CAN extension for time-triggered communication, the TTCAN bus, can also be simulated through a firmware upload on the simulator. The bus message set, the individual message specification and its timing requirements are downloaded to the hardware simulator through a PC connection. A software tool running on this PC records all timing information collected directly from the bus, on a bit time basis. This information can be used to evaluate specific message scheduling policies and network configurations for a message set. Results validation compares simulation values with those computed from an analytical model. Examples of simulations considering different bus window assignments and their effects over the message set schedulability are presented and discussed.

1 Introduction

Fieldbuses are low cost communication systems with increasing use on embedded systems (planes, ships e cars) and control and automation systems [CAN-CiA, 2002]. If those systems have to send messages with tight time requirements, their communication systems are classified as Real-Time Communication Systems (RTCS). Generally, on RTCS, a message can be classified as synchronous and not synchronous [ISO 11898, 1993]:. The first one has periodical occurrence, e.g., the result of a cyclic scan of sensors; the last one has an eventual behavior, not cyclic, and an alarm occurrence is a typical example of this type of message.

Network communication protocols can also be classified according to the adopted paradigm (event-triggered or time triggered) to match time requirements of RTCS messages. CAN [ISO 11898, 1993] and TTP/C [Kopetz and Grünsteidl, 1994] are examples of both paradigms respectively. Particularly interesting is the protocol presenting a proactive addition to both paradigms, the TTCAN protocol. This protocol assumes the advantages of both protocols while minimizing their disadvantages [ISO 11898-4, 2000].

Message scheduling is a very important technique applied to meet time requirements at RTCS. With protocols based on time-triggered paradigm, message scheduling problem can be solved through proprietary methods [Kopetz&Grünsteidl,1994], [Krüger,1995], [Pop,2000]. Based on heuristics, these methods solve scheduling problems on a feasible way not necessarily optimal [Coelho, 2002]. Also, schedulability testing [Tindell et al, 1995] is required to check in advance if specified requirements can be accomplished. The suitable choice of network configuration and message scheduling algorithms to provide proper operation of RTCS is an important design step. Simulation and analytical models solvers can play an important role at this step.

RTCS Simulation or analytical models can be solved through hardware or software tools. These tools allow the testing of several message scheduling policies or network protocol configurations to choose those that better meet message time requirements. This paper presents a CAN and TTCAN simulation tool built over DSP hardware with a PC interface. It supports simulation of TTCAN and CAN temporal behavior providing data about message scheduling policies, network configuration and their effects over message schedulability. The paper structure is as follows: Section 2 presents the essence, characteristics and operation of CAN and TTCAN protocols; Section 3 describes the Simulation System, its specification, hardware deployment and limitations as well the software module developed to overcome hardware limitations; Section 4 presents 2 two policies to map messages on TTCAN protocol windows; Section 5 presents analytical models of CAN and TTCAN temporal behaviors used to compare simulation results. At last, Section 6 discusses scenarios and experiments chosen to check the Simulator and discusses about network configurations to achieve better performances of RTCS.

2 CAN e TTCAN

CAN (Controller Area Network) standard is based on event-triggered protocol paradigm, presenting easy and low cost implementation and great scalability. Its bus access protocol solves message collisions based on message priority. Message retransmission, Fault tolerance and error control are also features of the CAN standard [ISO 11898, 1993].

TTCAN (*Time Triggered Controller Area Network*) is a CAN extension [ISO 11898-4, 2000] presenting the same CAN frame structure. Based on time triggered paradigm, the TTCAN medium access control establishes a message transmission pre-scheduling (TTCAN arbitration) where each node on the bus knows in advance the time interval (window time or slot time) it is allowed to transmit. Some free windows (CAN arbitration windows) are specifically reserved for asynchronous message transmission through CAN bus arbitration approach. A reference message (RF) is used to achieve global time synchronicity. Its reception provides a correct time reference to each node to transmit at the correct window based on its internal message list.

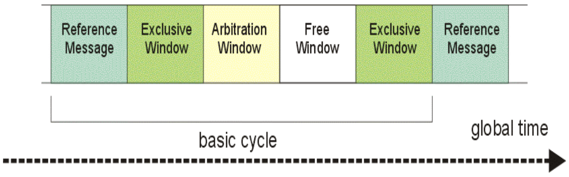

Figure 1 – Message transmission cycle

The BC (*basic cycle*) is the time interval comprising two consecutives RF messages. Figure 1 shows a set of windows and where each window corresponds to a sole message transmission. A synchronous message can be associated to an exclusive time window and an arbitration window is reserved to asynchronous message transmissions through CAN original arbitration process. A SM (*system matrix*) matrix is established from a periodic BC cycle set, considering the window ordering or mapping. At the system matrix, columns represent message windows and lines BC cycles.

3 The Simulation System

The simulation system comprises hardware and software modules. The hardware module was deployed specifically for CAN simulations, based on

Digital Signal Processing (DSP) and built from an educational kit [Hara et al, 2002]. It allows to observe, at bit time base, the temporal behavior of the CAN bus when submitted to a known message set workload. An analysis of this behavior provides insights to configure the CAN network as a RTCS. Also, it validates network configurations by comparing simulation results with those achieved from analytical model solutions [Ringler, 1998].

The hardware simulator comprises the educational kit, firmware and an expansion board card. The educational kit based on DSP TMS320F241 (Texas Instruments) has general use I/Os pins, ROM memory Flash of 8 KWords and RAM of 544 Words, configurable timers, interrupt controllers, analog-to-digital converter modules, CAN Module and one serial communication port to a microcomputer (very useful to transfer and program debug and also to network traffic monitoring). The firmware developed on a C language tool provides all basic operations of monitoring, port reading, task synchronization with a timer, message scheduling, queue management and delay accounting. The expansion board has a CAN transceiver to match interface and bus voltages, it supports bandwidth up to 500 kbps and connects through RJ-45 with 5 V, GND, CAN_H e CAN_L signals and with UTP cabling. It can act as terminal or gateway depending on resistive termination.

Nevertheless, for TTCAN simulations, the hardware module presented limitations due to processing power needed, high memory demand and bandwidth usage. To overcome these limitations (specially memory demand and TTCAN synchronization[1]), a software module on C language was developed to run on a personal computer (PC). The temporal behavior simulation of a TTCAN includes bit time monitoring of windows, BC cycles, reference messages, mapping of messages to windows and accounting of all message response times. The software module can consider any transmission bit rate just converting bit time to time period proportional to the real transmission bandwidth. The program runs on batch mode where a command file contains the message set (individual message characteristics and requirements) and system transmission characteristics (bandwidth, window size and number of windows per BC). The results are written on a data file enclosing the output traffic pattern and response time of each individual message.

At simulation mode, the program generates the message arrival pattern according to the previously message x window mapping provided to each node on the bus. It also generates all message instances of a synchronous message set during a time interval equals to the least common multiple (l.c.m.) of all individual message periods of the synchronous message set. This time interval is computed from the time instant where all messages are ready to transmit (critical instant) to the instant when all instances where already transmitted. For each bit time, all messages have their delay time counters monitored and a message ready to transmit is chosen to be transmitted at the next time window available.

4 Message Scheduling Algorithms for TTCAN

For a given message set, the assignment of its messages on TTCAN twindows determines the attendance or not of message time requirements. Our bibliographical review only found TTCAN window assignment policies based on proprietary programs, from there nor the nature neither the scheduling algorithms were revealed [Fonseca, 2001].

When using message priorities to assign TTCAN windows, the priority assignment algorithm plays a very important role to achieve schedulability. Message priorities once fixed should not be necessarily the same of the message source application. We have examined here two TTCAN window assignment policies, here denoted as PSPQ and TDMA, both based on fixed priority messages [Hara, 2003] and described at sections 4.1 e 4.2,

[1] TTCAN synchronizes at the beginning of a frame reception with a Start of Frame field (SOF).

respectively. Section 5 discusses some simulation results of both policies.

From now, we analyze the schedulability of a known fixed priority message set on a TTCAN system assuming no message preemption and a fixed size for one TTCAN window equals to the size of the biggest (lengthiest) message of the set.

4.1. Pre-scheduling based on preemptive queue (PSPQ)

We denote as PSPQ, the TTCAN window assignment algorithm to fixed priority messages. Based on a preemptive queue, this queue presents the same message order pattern of a preemptive system. A simulation of the message input traffic considers, for each new message arrival, the message placement at the queue based on its priority. Depending on it, some of the messages already in the queue can be displaced (preempted) and relocated to another position in the queue. This procedure repeats itself until the message arrival pattern completes, that is, the simulation time equals to the time interval determined by the least common multiple of all message periods of the message set under testing. At that point, the resulting queue presents a message queue similar to an output queue of a CAN network. The differences result from message frame sizes: CAN presents variable sizes while TTCAN always fixed size.

4.2. Time Division Multiple Access (TDMA)

The TDMA assigns each free time window according to the priority order of the messages, that is, the highest priority message is assigned to the first time window, the second highest to the second window and so on [Pop, 2000]. Figure 2 presents a BC cycle of a TDMA assignment: the time interval T_{JAN} is fixed, the RF message is always the first one at a BC cycle and the total interval T_{BC} equals $(n+1)\ T_{JAN}$. A message has a pre-allocated window that can remain free when no message instance occurs. Asynchronous messages should be considered for resource reservation. Idle windows represent reserved resources that were not used.

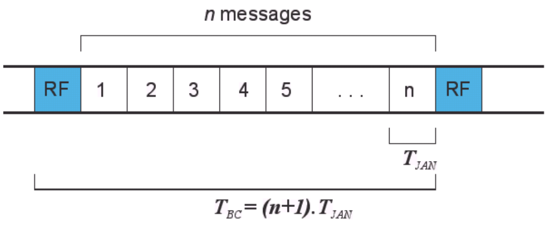

Figure 2 – TDMA assignment

5 Message schedulability on TTCAN networks

This section discusses some analytical models which solution results were compared with our simulation model solutions to validate and feedback the design of our simulation tool.

5.1. Analytical model of SCTR temporal behavior

The analytical model presented at [Tindell et al, 1995] allows a schedulability test of a fixed priority message set over a RTCS. Aiming pre-run time guarantees, results of such test are important at RTCS design to accomplish message time requirements.

It is out of the purpose of this paper to present details of the model, but some of its concepts should be shortly introduced for the understanding of the simulator evaluation. For the sake of clarity, assuming Tindell's model for fixed priority messages, statistically independent messages, message deadlines equal or lower than their occurrence periods, Equation (1) allows the computation of the worst case of response time *q* of a message *i* given:

The message set M: {m_1, m_2 ,... m_i,... m_n};

a) The number *n* of elements of the message set *M*;
b) The message characteristics $m_{i:}\{C_i, T_i, D_i, ...\}$, where *C* is the message size (as the necessary time to transmit it without preemption at the nominal network bit rate), *T* the occurrence period and *D* its deadline;
c) The individual and unique fixed priority of the messages;
d) The network temporal behavior.

B_i represents the blocking time caused by priority inversion, that is, the time interval one message with priority *i* can be blocked by the transmission of a message *j* with lower priority than *i*. The delay inflicted by the set *hp(i)* of messages *j* of higher priority than *i* already in the transmission queue is given by the summation below where Cj represents the size of the message *j*; T_j is the message *j* period and p_{bit} the bit propagation delay on network physical medium.

$$q_i = B_i + \sum_{j \in hp(i)} \left\lceil \frac{q_i + p_{bit}}{T_j} \right\rceil \cdot C_j \qquad (1)$$

The schedulability test of *M* checks for each message m_i of *M* ,$\forall$ *i*, $0 < i < n$. If $q_i < D_i$ then the message set *M* is schedulable[2].

5.2. Schedulability test of a message set M over TTCAN with PSPQ

Equation (1) can be refined to include the impact of TTCAN access medium protocol and PSPQ on the schedulability of a fixed priority message set. As seen at section 3.1., when using PSPQ, the message arrival pattern is similar to the message transmission order on a CAN network. Therefore, the analytical model is similar to that presented by [Tindell et al, 1995] to CAN protocols. Thus, the message *i* response time given by *Wi* is:

$$W_i^{k+1} = C_{ref} + T_{JAN} \cdot \left(1 + \sum_{j \in hp(i)} \left\lceil \frac{W_i^k}{T_j} \right\rceil \cdot T_{JAN} \right) \qquad (2)$$

All time windows have fixed size. Therefore, B_i from Equation (1) presents an blocking upper bound equal to the window T_{JAN}. The RF message impact over a message *i* at each BC cycle is denoted as C_{ref}. The model assumes the message set M as entirely pre-assigned to a unique BC cycle which cycle time comprises a message arrival pattern equals to the Least Common Multiple (l.c.m.) of the message periods. This assumption implies the arrival of only one RF message. If W_i^{k+1} converges to W_i^k, the satisfaction of (3) indicates the schedulability of M, i.e., all messages will match their deadline requirement.

$$\forall i \mid \; 0 < i < n, \qquad W_i < D_i \qquad (3)$$

5.3. Schedulability test of a message set *M* over TTCAN with TDMA

The M set schedulability test over TTCAN using TDMA assignment policy can be verified by an utilization test [Liu&Layland, 1973]. The TTCAN utilization U_T can be computed as (4):

$$U_T = \frac{T_{JAN} \cdot \left(\left\lceil \frac{X}{(n+1)T_{JAN}} \right\rceil + \left\lceil \frac{X}{T_1} \right\rceil + \left\lceil \frac{X}{T_2} \right\rceil + \ldots + \left\lceil \frac{X}{T_n} \right\rceil \right)}{X}$$

where

$$\frac{T_{JAN} \cdot \left\lceil \frac{X}{T_i} \right\rceil}{X} \quad \text{and} \quad \frac{T_{JAN} \cdot \left\lceil \frac{X}{(n+1) \cdot T_{JAN}} \right\rceil}{X}$$

represent, respectively, the message *i* and the RF message utilization. The necessary requirement to schedulability is $U_T \leq 1$*(5).* With (4) at (5):

$$\frac{T_{JAN} \left\lceil \frac{X}{(n+1) \cdot T_{JAN}} \right\rceil + T_{JAN} \cdot \sum_{i=1}^{n} \left\lceil \frac{X}{T_i} \right\rceil}{X} \leq 1 \qquad (6)$$

where

U_T transmission medium utilization;

T_{JAN} time window interval;

X least common multiple of all message periods including the RF message.

The utilization test is only a necessary test, i.e.: if approved nothing can be said about schedulability, but if not, the message set is not schedulable.

6 Experiments

The scenarios chosen to perform the simulation experiments consider: two message sets, fixed priority scheduling algorithms, two window assignments, bit rate, window size. The message set (with 6 messages and denoted as set *A*), was chosen from [Hansson et al, 2000] and shown at Table 1. A message set *B* includes one more message. The column q_i presents results obtained from an analytical model with the indicated message priority assignment. Assuming an ideal preemptive priority scheduler, the set *A* of Table 1 is schedulable and set *B* is not ($q_7 > D_7$).

[2] We check here only the accomplishment of the deadline requirement.

Table 1 – message sets applied at the experiments.

		Bandwidth 25 kbps					
Order	Set id	Message i	Priority	T_i	D_i	C_i	q_i
1	A, B	Op. 1	1	80	80	5,4	10,8
2	A, B	ABS-1	2	40	40	5,4	16,2
3	A, B	ABS-2	3	40	40	5,4	21,6
4	A, B	ABS-3	4	40	40	5,4	27,0
5	A, B	ABS-4	5	40	40	5,4	32,4
6	A, B	Op. 2	6	150	150	5,4	37,8
7	B	Op. 3	7	30	30	5,4	49,9

Table 2 presents CAN simulation results of sets A and B with both models: simulation and analytical (Equation 2). The hardware simulation of the experiment number 3 reveals the pessimism of the analytical model result.

Table 2 – CAN experiment results

		Bandwidth 25 Kbps		
Exp Id	Analytical model test result	Simulation model test result	Message set	Message priority Order
1	Schedulable	Schedulable	A	1,2,3,4,5,6
2	Non-schedulable	Non-schedulable	B	1,2,3,4,5,6,7
3	Non-schedulable	Schedulable	B	7,1,2,3,4,5,6
4	Non-schedulable	Non-schedulable	B	1,2,3,4,5,7,6

Time synchronization of the hardware simulator happens only after frame reception. Because CAN frames have variable size due to bit stuffing, synchronization causes jitter on the time base. To circumvent this effect, TTCAN simulations execute on a software module. All TTCAN network terminals should begin their first BC cycle synchronized and with delay and period counters set to zero. We assume error control only at application level, implying on message retransmission only at a free window (not previously assigned) through CAN bus arbitration. The TTCAN operation under PSPQ considers window to message mapping obtained from the final preemptive queue simulation explained at section 3.1. First, a scheduler module scans at each program loop (equivalent to a bit transmission time) if there is any message arrival. The loop execution also increases the delay counter of each message at the transmission queue and places the pending messages at the preemptive queue. Before a window transmission starts, the highest priority message at the pending message queue is mapped to the current time window (exactly as the CAN bus arbitration process would done). This procedure is repeated until the equivalent of the l.c.m. of all message periods of the message set under test. The transmission pattern achieved is similar to that transmitted by CAN, unless for the frame sizes. Table 3 presents the results of TTCAN simulations.

Table 3 – TTCAN experiments results

BW (kbps)	Test goal	WDW assignm.	WDW size (ms)	D Miss	SM col.	# of BC cycles
50	Analysis of BC cycle impact	PSPQ	3,28	-	3	2
50	Analysis of BC cycle impact	PSPQ	3,28	-	6	1
50	Temporal Redundancy	PSPQ	3,28	-	12	1
50	Analysis of BC cycle impact	TDMA	3,28	Yes	3	2
50	Analysis of BC cycle impact	TDMA	3,28	-	6	1
25	TTCAN using CAN	TDMA	6,56	Yes	6	1
50	Temporal Redundancy	TDMA	3,28	Yes	12	1
62,5	Temporal Redundancy	TDMA	2,624	-	12	1

The results show the impact of BC cycles on the message set schedulability: the overhead imposed by the number of RF messages is directly related to the BC cycle size. Also, the lesser the time window size better the network performance.

7 Final Comments

Some interesting insights about TTCAN configuration were learned from the simulations: e.g., once a message set presents itself as non-schedulable at an specific network configuration, small changes (decrease) on window interval results on better bandwidth use. Therefore, size choice of window should be carefully done. A first approach suggests the size of the biggest (lengthiest) message, once preemption is not considered (fragmentation). Also, testing of assignment policies for window size and message priorities can be conducted through simulation use, for example, as the heuristics used to develop PSPQ. Temporal Redundancy reveals itself very interesting while minimizing bandwidth idleness. Differently from TTP/C, the TTCAN allows free window (non-allocated) use through CAN arbitration. The impact of these allocations can determine the schedulability of a message set on a heavy traffic network.

The scenarios of Table 3 allowed tests of several parameter configurations but we are still collecting results. From this first step, it is clear for us the importance of simulations and its result validation to establish new approaches (directives) to TTCAN and CAN configuration aiming to accomplish real time requirements.

Considering the educational use of the simulation tool, CAN and TTCAN protocol features are made easy to understand, especially to smooth analytical models comprehension.

8 References

[CAN-CiA, 2002] CAN-CiA. "**Controller Area Network (CAN), an Overview**", http://www.can-cia.org/can/, 2002.

[Coelho, 2002] J. M. A. Coelho. **"Aprendizado de Heurísticas para o Escalonamento de Sistemas de Tempo Real"**, WTR – 20° SBRC – Brasil, 2002.

[Fonseca, 2001] J. Fonseca, F. Coutinho, J. Barreiros. "**Scheduling for a TTCAN network with a stochastic optimization algorithm**", proceedings of 4th FET'2001, IFAC Conference on Fieldbus Systems and their Applications, Nancy, France, Nov. 2001.

[Liu&Layland, 1973] C. L. Liu and J. W. Layland: **Scheduling algorithms for multiprogramming in a hard-real-time environment.** Journal of the ACM, 20(1), 1973.

[Hansson et al, 2000] H. Hansson, C. Norström, S. Punnekkat. "**Integrating Reliability and Timing Analysis of CAN-based Systems**", IEEE Workshop on Factory Communication Systems (WFCS-2000), Porto, Portugal, September 2000.

[Hara et al, 2002] A. Hara, E. Wagner, K. Fonseca, L. Scandelari, E. Nisenbaum. **"Um Sistema Didático (Hardware e Firmware) para simular redes tipo CAN"**, WTR – 20° SBRC – Brasil, 2002.

[Hara, 2003] A. Hara. **"Uma Proposta de Heurística para Configuração de TTCAN como Sistema de Comunicação Tempo Real"**, MSc. Thesis 271, CEFET-PR, CPGEI, Curitiba, Brasil, 2003.

[ISO 11898, 1993] ISO 11898. "**Road Vehicles – Interchange of Digital Information - Controller Area Network (CAN) for High-Speed Communication**", Technical Committee ISO/TC 22, Road vehicles, Subcommittee SC3, Electrical and Electronics equipment, Nov. 1993.

[ISO 11898-4, 2000] ISO 11898-4. "**Road Vehicles – Controller Area Network (CAN) – Part 4: Time Triggered Communication**", Technical Committee ISO/TC 22, Road vehicles, Subcommittee SC3, Electrical and electronics equipment, Dec. 2000.

[Kopetz e Grünsteidl, 1994] H. Kopetz, G. Grünsteidl. "**TTP – A Protocol for Fault-Tolerant Real Time Systems**", IEEE Computer, pp. 14-23, January 1994.

[Krüger, 1995] A. Krüger. "**Performance Constraints for a Time-Triggered Communication Controller**", citeseer.nj.nec.com/163555.html, European Research Seminar on Advances in Distributed Systems, Alpe d'Huez, France, April 1995.

[Pop, 2000] P. Pop. "**Scheduling and Communication Synthesis for Distributed Real Time Systems**", ISBN 91-7219-776-5, ISSN 0280-7971, Institute of Technology - Linköping University, Linköping, Sweden, 2000.

[Ringler, 1998] T. Ringler, J. Steiner, R. Belschner, B. Hedenetz. **"Increasing Systems Safety for by-wire Applications in Vehicles by Using a Time Triggered Architecture"**, SAFECOMP, 17th Int. Conf. on Computer Safty, Reliability and Security, Heidelberg, 1998.

[Tindell et al, 1995] K. Tindell, A. Burns, A. Wellings. "**Calculating Controller Area Network (CAN) Message Response Times**", Control Engineering Practice 3 (8): 1163-1169, 1995.

Artur Tsuguiyoshi Hara, MSc. Electrical Engineering (CEFET-PR, 2003)
CEFET-PR
Av. Sete de Setembro 3165 Curitiba-PR, Brasil
+55413104695
+55413104683
Artur@cpgei.efetpr.br
www.cpgei.cefetpr.br

Keiko V.O. Fonseca, Dr. Electrical Engineering (Universidade Federal de Santa Catarina,1997)
CEFET-PR
keiko@cpgei.cefetpr.br

Luciano Scandelari, PhD Physics (Université Joseph Fourier, France, 1999)
CEFET-PR
luciano@cefetpr.br

Speaker: Artur Tsuguiyoshi Hara
MSc. Electrical Engineering, 2003 at CEFET-PR, Curitiba-PR, Brazil,
teacher at CEFET-PR, Pato Branco campus, Paraná State, Brazil