Time Triggered CAN, Implementations, Development and Testing Tools

Chris Quigley, Ben Pope, James Finney, Richard T. McLaughlin
Warwick Control Technologies

ABSTRACT

The Controller Area Network (CAN) has seen enormous success in automotive body and powertrain control systems, and in industrial automation systems using higher layer protocols such as DeviceNet and CANopen. Now, the CAN standard ISO11898 are being extended to Time Triggered CAN (TTCAN) to address the safety critical needs of first generation drive-by-wire systems. However, their successful development depends upon the availability of silicon and software support, and appropriate development & analysis tools.

This paper outlines some of the exciting developments at Warwick Control Technologies which includes:-

a) An implementation that can run deterministically at 60% bus loading at 1Mbit/s and 70% at 500Kbit/s (almost twice that of traditional automotive CAN systems),
b) Interoperability testing with the Bosch
c) Rapid control prototyping for a TTCAN Embedded Node, and
d) TTCAN transmitter/fault injector tool.

Introduction

Of the currently available open standard network protocols, CAN has become the most prominent across the world's automotive industry. The probable reasons for this have been its huge support from major semiconductor manufacturers, tool suppliers and automotive OEMs. Now, nearly all automotive OEMs have products available with CAN or are intending to develop with CAN in the near future. However, industrial automation applications using DeviceNet and CANopen were amongst the first applications of CAN technology.

This paper focuses on the research and development carried out by Warwick Control Technologies on TTCAN technology which includes:-

- Atmel T89c51cc01 microcontroller implementation providing higher bus loading
- Interoperability work with the Bosch evaluation chip
- TTCAN fault injection
- Configuration tool for TTCAN embedded nodes

.

The *CAN communications protocol* specifies the method by which data is passed between communicating devices on a CAN bus. It conforms to ISO's Open System Interconnection (OSI) model, which is a seven-layer description of a telecommunications network standard.

In fact the CAN protocol can be described by the lowest two layers of the OSI model – the Data Link Layer and the Physical Layer. The Application layer protocols can be proprietary schemes developed by individual CAN users, or one of the emerging standards used within particular industries. A common application layer standard used in the process-control/manufacturing field is DeviceNet, which is especially suited to the networking of PLCs and intelligent transducers. In the automotive industry most manufacturers use their own proprietary standard. In the utility vehicles industry, J1939 has become the popular application layer. This paper assume the reader's prior knowledge of the CAN Physical, Data Link, and Physical Layers.

Time Triggered CAN is standardised under ISO-11898-4 which covers four main parts:

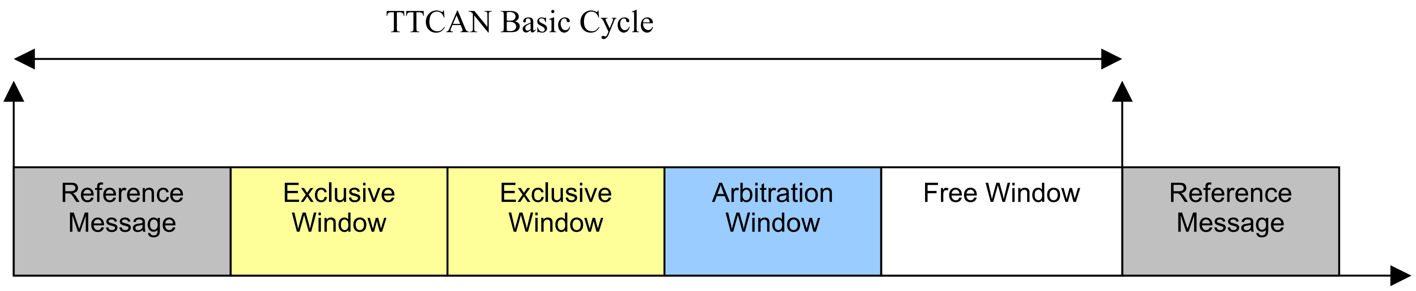


Figure 1: An example of the Basic Cycle of time triggered CAN communication

- Time triggered execution of CAN with a central Time Master periodic messages, event triggered messages and Time Master Synchronisation.
- Fault tolerance of this Time Master.
- Drift correction and Global Time.
- Event synchronised - time triggered communication

TTCAN is now available in silicon implementations from the following manufacturers:-

- Atmel
- Infineon
- Microchip
- NEC

The main characteristics of TTCAN is that bus access is controlled via a Time Division Multiplexed Access (TDMA) like method using a regularly repeating cycle of time called the *Basic Cycle* (see figure 1). The *Basic Cycle* is divided into a fixed number of time windows (i.e. fixed at design time) which can be a mixture of any one of four types; *Reference Message*, *Exclusive Window*, *Arbitration Window* and *Free Window*.

Reference Message:

This is sent by the time master control unit (global time master) and controls the timing of the Basic Cycle. The Reference Message signifies the start of the Basic Cycle. CAN communication is initiated by a Reference Message. The global time can be sent in four data bytes leaving the remaining four data bytes available for general data transfer.

- The identifier of the Reference Message is determined by the system designer.
- All but the three least significant bits of this identifier characterise the message as a reference message.
- The three least significant bits distinguish between different potential time masters.
- These three LSBs constitute the time master priority.
- There are up to 8 potential time masters in a TTCAN network.

Exclusive Window:

This is a time slice long enough to accommodate the message and data to be transmitted. The *Exclusive Window* is reserved for one particular CAN message only.

Arbitration Window (optional):

In an Arbitration Window a number of nodes may attempt to transmit a message. Therefore the nodes that may contend for bus access during the *Arbitration Window* may do so by the usual non-destructive bitwise arbitration method. Thus the message with the lowest CAN identifier will win arbitration. With normal CAN systems, nodes losing arbitration will attempt to retransmit the message. This is disabled in TTCAN since a retransmission would upset the remainder of the operation of the Basic Cycle.

Free Window (optional):

A Free Window is reserved for future expansion of the TTCAN system. Therefore further nodes can be added at a later date.

All nodes communicating on a bus, which is to be time triggered, must have the retransmission of message that has lost arbitration disabled.

Synchronisation of Cycle Time

Cycle time is resynchronised with every reference message. Accuracy of the starting point depends on width of frame synchronisation pulse and on measurement capability of receiving nodes.

In level 1 the starting point jitter is about one bit time. In level 2 the starting point jitter can be reduced to about one time quanta, if the local time counter has a sufficiently high resolution. The effect of run times is constant and can therefore be neglected for most synchronisation items.

Potential Time Masters

There can be up to 8 potential time masters in a TTCAN network. At a given time only one of them can be the current time master. Each of the potential time masters has a 3-bit time master priority. Within a TTCAN network the time master priorities of different time masters have to be different. The time master priorities have to be defined offline. If a potential time master sends a reference message the 3 LSBs of the identifier of the reference message are given by the time master priority. In a fault free TTCAN network after initialisation the node with the highest time master priority will be the time master.

Main Advantages of TTCAN

- Provides a time deterministic method of communication based on existing technology.
- Since it is based on existing technology, expertise for TTCAN will be based on a company's existing CAN knowledge and experiences.
- Development tools originally aimed at CAN can be used for the analysis of data on TTCAN. Direct monitoring of TTCAN with existing CAN bus analysis tools is possible since TTCAN frames are exactly the same as for CAN. However, this is not necessarily possible if transmitting on the TTCAN bus, since current tools will almost definitely interfere with the message schedule.
- Thus the transition from CAN to TTCAN systems will not require a huge financial outlay.
- Increases data throughput by enabling the bus to be run at high loading.

Atmel TTCAN Implementation on T89c51cc01

Silicon Features:

TTCAN is available technology now for the first generation of drive-by-wire systems. The Atmel CANary CAN controller supports the implementation to TTCAN Level 1 currently in their T89C51CC01/02/03/04 8051 based microcontrollers. The main TTCAN Level 1 silicon support features are summarised as:

- **Disable Retransmission**: For any message object that loses arbitration, or is corrupted by an error, an attempt is made to retransmit that message object. The CANary CAN controller allows this retransmission of message objects to be disabled.
- **SOF and EOF Timestamp Capture**: The CANary CAN controller allows for the capturing of a timestamp at the beginning or end of a CAN frame. Capturing of either SOF and EOF allows for network synchronisation.

Due to the limitations of the CANary controller, Exclusive window implementation is described in subsequent sections only. The implementation of Arbitration windows is not recommended. Full reasons are explained in a later section. Appendix C shows the contents and details of these CANary registers specific to TTCAN operation.

Summary of Memory Usage:

The Memory Requirements Table in Appendix A, illustrates the relationship between the number of cycles, the number of widows in each cycle, and their effect on the memory requirements of the Atmel development unit. For example if 8 Cycles are required, and there are 16 windows per cycle, it can be seen the memory requirements will not be overtaxed. Only 490 Bytes of XRAM are required, and the maximum usage before the need of external memory is 906 Bytes.

The main code base without any messages:

- ~ 2 K bytes code

  - Time Master/Potential Time Master 2100 bytes
  - Slave 1850 bytes
- 66 bytes data
- 65 bytes xdata

Summary of CANary Controller Usage:

The CANary CAN controller is the on-chip peripheral used for CAN communications. All CAN chips have 15 Message Filtering Objects for on chip message filtering. This removes demand from the microcontroller. For TTCAN communications, each of the CAN communication objects is used as below:

Object 0:

- Transmission of Reference Message

Object 1:

- Reception of Reference Message

Objects 2 to 14:

- Msg *A* to *M* respectively – used as Full CAN Objects
- Can be used for Transmission or Reception of messages

Reference Message Implementation:

CAN **Object 0** is used for the generation of the TTCAN Reference Message as mentioned previously. The most significant bits of the CAN Identifier field must match. This is the 8 most significant bits of an eleven bit CAN Identifier or the 26 most significant bits of a twenty nine bit CAN Identifier. The 3 least significant bits of the CAN Identifier, are effectively "don't care" bits and are only necessary upon initialisation and re-initialisation situations and required for Time Master Priority.

CAN **Object 1** is used for the Reference Message reception in Potential Time Master and Slave devices. When a Reference Message is received in Object 1, an interrupt is generated. The interrupt service routine then instigates the calculation of Cycle Error so that the node can synchronise to the TTCAN Master.

Time Master Failure Fault Confinement:

If the current Time Master fails, the next Potential Time Master takes over automatically. This is achieved via the Init_Ref_Offset mechanism described previously. However, for the CANary implementation Init_Ref_Offset must be set to:

TM_Priority * 8

for hand over to happen smoothly. This is due to clock drift and internal delays and jitter in the CANary CAN Controller.

Therefore the Init_Ref_Offset settings for different Time Master Priority nodes are shown in table 1.

Time Master Priority	Init_Ref_Offset
0	0
1	8
2	16
3	24
4	32
5	40
6	48
7	56

Table 1: Time Master Priority and Init_Ref_Offset

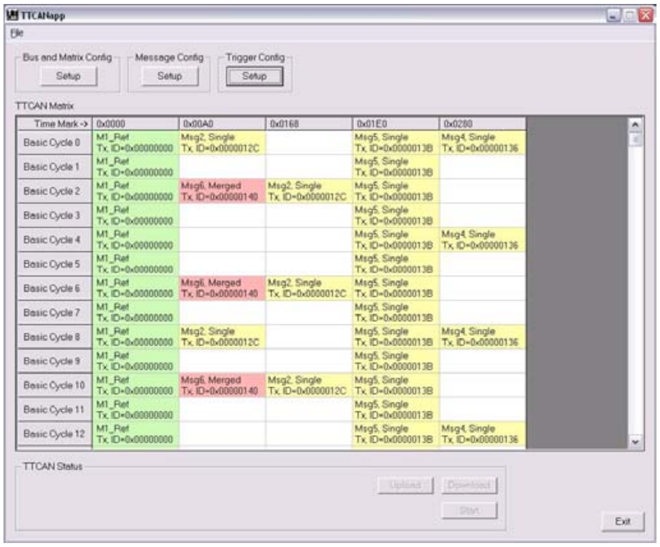
Analysis of TTCAN Bus loading of the Atmel T89c51cc01 Implementation

Referring to the Performance Table in Appendix A, It can be seen that the bus load capabilities of TTCAN, is reliant on the internal clock of the microcontroller, and the data bus speed. For example, at the typical clock rate of 16 MHz, and at the typical automotive bus speed of 500Kbps, the bench testing of the Atmel TTCAN development board showed no significant message latencies until the bus load reached 67%.

This is a significant improvement over the traditional contention based messaging method of CAN. Past research has shown that contention based CAN messaging experiences significant message latencies at approximately 35%.

TTCAN Drivers Configuration, Interoperability Testing and Fault Injection

A TTCAN Application Configuration Tool was developed to ease the design times of message scheduling. It graphically enables a designer to configure the communication matrix and allow the scheduling of each message window (Reference Message, Exclusive Window, Arbitration Window, Free Window). This tool can configure the Master, Potential Time Masters (PTM) or Slaves of each TTCAN controller. Figure 2 shows an example set-up for a TTCAN node. Once the graphic is completed, C code header files are generated, and then downloaded to the unit under test to complete the matrix table configuration.



TTCAN Matrix

Time Mark ->	0x0000	0x00A0	0x0168	0x01E0	0x0280
Basic Cycle 0	M1_Ref Tx, ID=0x00000000	Msg2, Single Tx, ID=0x0000012C		Msg5, Single Tx, ID=0x0000013B	Msg4, Single Tx, ID=0x00000136
Basic Cycle 1	M1_Ref Tx, ID=0x00000000			Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 2	M1_Ref Tx, ID=0x00000000	Msg6, Merged Tx, ID=0x00000140	Msg2, Single Tx, ID=0x0000012C	Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 3	M1_Ref Tx, ID=0x00000000			Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 4	M1_Ref Tx, ID=0x00000000			Msg5, Single Tx, ID=0x0000013B	Msg4, Single Tx, ID=0x00000136
Basic Cycle 5	M1_Ref Tx, ID=0x00000000			Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 6	M1_Ref Tx, ID=0x00000000	Msg6, Merged Tx, ID=0x00000140	Msg2, Single Tx, ID=0x0000012C	Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 7	M1_Ref Tx, ID=0x00000000			Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 8	M1_Ref Tx, ID=0x00000000	Msg2, Single Tx, ID=0x0000012C		Msg5, Single Tx, ID=0x0000013B	Msg4, Single Tx, ID=0x00000136
Basic Cycle 9	M1_Ref Tx, ID=0x00000000			Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 10	M1_Ref Tx, ID=0x00000000	Msg6, Merged Tx, ID=0x00000140	Msg2, Single Tx, ID=0x0000012C	Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 11	M1_Ref Tx, ID=0x00000000			Msg5, Single Tx, ID=0x0000013B	
Basic Cycle 12	M1_Ref Tx, ID=0x00000000			Msg5, Single Tx, ID=0x0000013B	Msg4, Single Tx, ID=0x00000136

Figure 2: GUI for driver set-up

The tool can also download a configuration to an Embedded Node, from the host PC via serial link such as RS232. See figure 3 for an example configuration. With this type of set up, it is possible to simply simulate a TTCAN node. Alternatively fault injection can be achieved, e.g.:-

- Adjust Time Marks to incorrect values (e.g. advance or retard the Time Mark)
- Adjust DLC to incorrect values and see if a receiving node detects the error
- Adjust CAN identifiers

It is also possible to upload the Unit Under Test (UUT) existing configuration into the TTCAN Application Configuration Tool.

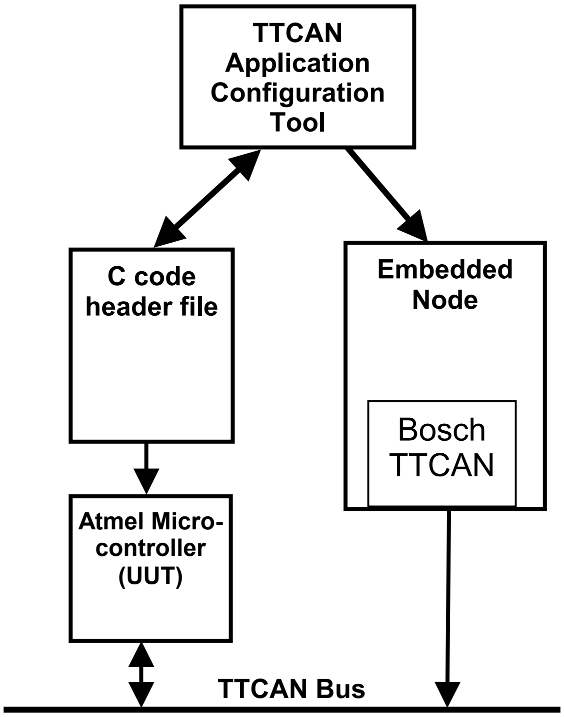

TTCAN Application Configuration Tool
C code header file
Embedded Node
Bosch TTCAN
Atmel Micro-controller (UUT)
TTCAN Bus

Figure 3: TTCAN Matrix Configuration set-up, Fault Injection, and Interoperability Testing with a Bosch Evaluation Chip

Implementations of TTCAN

In this section possible applications and implementations of TTCAN technology are mentioned with particular regard to the Atmel T89c51cc01 with CANary CAN controller.

High Speed Auxiliary TTCAN System

High Speed Auxillary CAN systems internally to the vehicle are often used to pass signals to one other system and are not usually part of the main vehicle CAN bus. This type of configuration is illustrated in Figure 4. Here the ESP and the ABS are required update every 500 microsecond. This would overtax the bus loading of the Powertrain CAN bus.

The application here needs a closed loop cycle time of down to 500us and this is not possible with CAN. With TTCAN this is possible at 500 Kbaud and 1 Mbaud. By implementing these as TTCAN, the following benefits can be found:

- Increased message determinism
- Which leads to higher system integrity
- Easily implement message watch dogs so that if a message does not arrive in a timely manner, then a failsafe can be activated

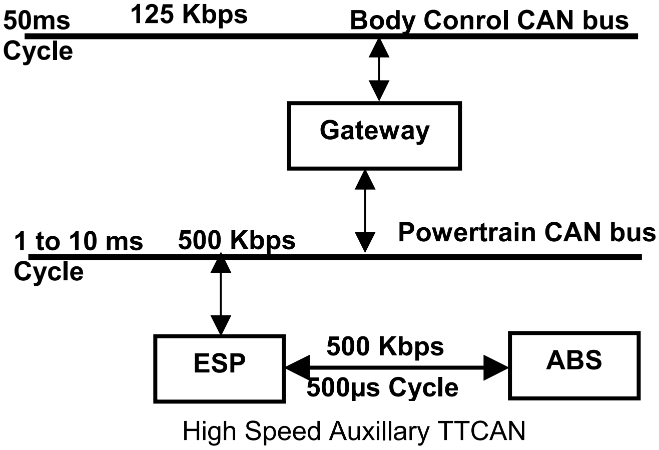


Figure 4: Example of High Speed Auxillary TTCAN

Example Applications of TTCAN CANary

Figure 5 shows two possible applications of the CANary TTCAN implementation; *Intelligent TTCAN Transducer* node and *TTCAN Communications Controller* for another host.

In the case of the *Intelligent TTCAN Transducer*, the complete Transducer handling and TTCAN communications is carried out on a single Atmel microcontroller such as the T89c51cc01.

In the case of the *TTCAN Communications Controller*, the Atmel microcontroller is used to solely deal with the TTCAN communications and application level processing is carried out on a host microcontroller.

Integration of OSI Layer 7 Protocols

OSI layer 7 protocols such as SAE J1939 can be easily integrated into the Time Triggered CAN protocol and TTCAN drivers such as the ones described in this paper. For a protocol such as SAE J1939, this brings benefits such as:

- Higher achieved bus loading
- Increased message determinism
- Higher system integrity due to this increased determinism
- Plug and play functionality can be implemented by leaving some spare slots

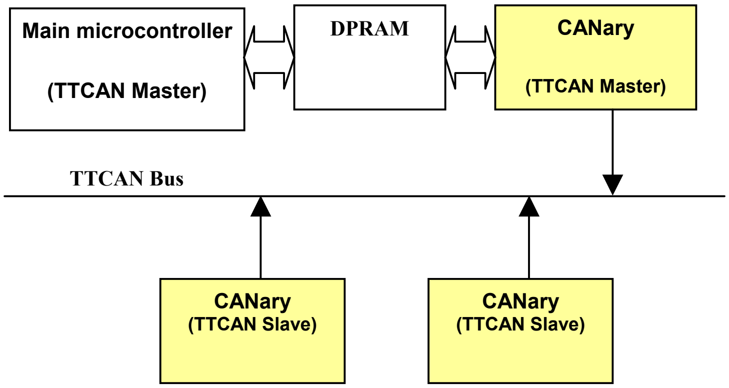


Figure 5: Possible applications of the Atmel CANary TTCAN implementation

Summary and Conclusion:

Future automotive control systems will utilise data communication networks in far more safety critical applications such as drive-by-wire. TTCAN has been developed to address the needs of these applications. Whilst TTCAN is limited in bandwidth and does not provide for all of the needs of such applications, it is very useful as the enabling technology in the first generation of drive-by-wire systems. TTCAN only requires a small amount of additional engineering knowledge on top of that already available from the CAN community and allows the same development tools as used for CAN to be used for message monitoring (e.g. CAN bus analyser software).

It has been shown that the Atmel CANary CAN controller with resides in the T89c51cc01 /02 /03 /04 microcontrollers can be used to implement the

TTCAN protocol. The most interesting and useful outcome of this implementation is that it can effectively double the usable bandwidth of CAN to a useable 70% bus loading. Automotive control system designers are limited to 1000 Kbaud by the CAN protocol, but usually limit their usable bit rate to 500 Kbaud for EMC reasons. Use of TTCAN effectively doubles the bandwidth whilst providing reliable deterministic communication latencies.

However, there are limitations to the implementation and therefore the following suggestions to its use are made:

- If the Atmel CANary is to be used in a full TTCAN bus implementation, it is recommended that it is used it as a TTCAN slave and only transmits messages in Exclusive windows.
- If the CANary is to be used as a Potential Time Master (TM0 to TM7) then it is recommended that its failure operation be limited. If when in use it is the Time Master and is therefore responsible for the transmission of the Reference Message, it is recommended that after a failure it should not be allowed to reconnect to the TTCAN bus without a system reset.
- It is recommended that a CANary node should not have the responsibility of transmission in an Arbitration Window. However, Arbitration Window reception is recommended.
- The bit rate of the CANary TTCAN implementation is limited by crystal oscillator used. At 500 Kbaud 70% loading can be achieved.

It is recommended that future work be performed to determine the performance of other TTCAN silicon as has been done with the Atmel chip here.

REFERENCES

1. ISO-11898 Part 4
2. Fuhrer T et al (2000); “Time Triggered Communication on CAN (Time Triggered CAN-TTCAN)”, Proceedings of the 7th International CAN Conference, Amsterdam, 24th and 25th October 2000.
3. Hartwich et al (2000); “CAN Network with Time Triggered Communication”, Proceedings of the 7th International CAN Conference, Amsterdam, 24th and 25th October 2000.
4. McLaughlin & Quigley (2004); “Analysis and Diagnostics of Time Triggered CAN (TTCAN) Systems”, SAE Congress paper 2004-01-0201.

Each of the authors are from:

Warwick Control Technologies Limited
Prodrive Warwick
Oldwich Lane East
Fen End, Warwickshire
CV8 1NR
United Kingdom

Richard T. McLaughlin, B.Sc., M.Sc., CEng MIEE
richard@warwickcontrol.com

Chris Quigley, B.Eng., M.Sc.
chris@warwickcontrol.com

Ben Pope, M.Eng.
ben@warwickcontrol.com

James Finney, B.Eng.
james@warwickcontrol.com

Appendix A – T89C51CC01 Time Triggered Communication Performance Metrics Summary

Memory Requirements Table

Absolute xdata requirement for Matrix

		Windows								
		1	2	4	8	16	32	64	128	256
Cycles	1	58	63	73	93	133	213	373	693	1333
	2	64	72	88	120	184	312	568	1080	2104
	4	76	90	118	174	286	510	958	1854	3646
	8	100	126	178	282	490	906	1738	3402	6730
	16	148	198	298	498	898	1698	3298	6498	12898
	32	244	342	538	930	1714	3282	6418	12690	25234
	64	436	630	1018	0	3346	6450	12658	25074	49906

🟧 CC03 or External XRAM required

data required
80

Performance Table

The statistics below show the variety of problems associated with Window Length and baud rate. Typically, an increase in Baud Rate requires an increase in Window Length when using the CANary as a Master or Potential Time Master.

Crystal Frequency	Baud rate	Time Master Microcontroller Loading	Slave /Potential Time Master Microcontroller Loading	Max CAN Bus Loading	Suggested Window length (Bit time)	Max. Windows per Basic Cycle
12 MHz	50 Kbps	5.9%	4.8%	79%	165	9
	100 Kbps	11.8%	10.3%	79%	165	19
	125 Kbps	14.5%	12.9%	79%	165	24
	250 Kbps	26.7%	23.6%	73%	180	45
	500 Kbps	45.7%	40.5%	62%	210	78
	1000 Kbps		Not Possible			
16 MHz	50 Kbps	4.2%	4.2%	79%	165	7
	100 Kbps	8.5%	8.5%	79%	165	14
	125 Kbps	10.6%	10.6%	79%	165	18
	250 Kbps	20.0%	20.0%	75%	175	35
	500 Kbps	35.9%	35.9%	67%	195	63
	1000 Kbps	59.6%	59.6%	56%	235	104
20 MHz	50 Kbps	4.8%	4.2%	79%	165	5
	100 Kbps	9.7%	8.5%	79%	165	11
	125 Kbps	12.1%	10.6%	79%	165	14
	250 Kbps	22.9%	20.0%	75%	175	28
	500 Kbps	43.2%	37.8%	71%	185	53
	1000 Kbps	80.0%	70.0%	60%	220	89

Bus loading is calculated by: bus load = 131bits/Suggested Window Length x 100%

It is assumed that 100% loading would be with TTCAN windows of 131 bits per 29 Bit CAN ID with 8 DLC frame.