iCC 1997

4th international CAN Conference

in Berlin (Germany)

Sponsored by

Motorola Semiconductor
NEC Electronics (Europe)
Siemens Semiconductors

Organized by

CAN in Automation (CiA)
international users and manufacturers group
Am Weichselgarten 26
D-91058 Erlangen
Phone +49-9131-69086-0
Fax +49-9131-69086-79
Email:headquarters@can-cia.de
URL: http://www.can-cia.de

Peter Hank
Philips Semiconductors, Hamburg

PeliCAN: A New CAN Controller Supporting Diagnosis and System Optimization

Abstract

System Optimization, Diagnosis and Maintenance are becoming more and more important for improving the performance of CAN Bus systems. Therefore, modern CAN Controller architectures have to provide special functions supporting these requirements.

This paper describes how dedicated features of a new CAN Controller can be used for this purpose. These functions also support an interesting approach for "Plug & Play" solutions with future CAN systems.

1. Introduction

Philips continues the development of CAN devices with the new Stand-alone CAN controller SJA1000. It is a successor of the well-known PCA82C200 [1], which already runs in hundreds of applications. Customers who have based their applications on the PCA82C200 are further supported with the new CAN controller, which is both hardware and software compatible.

In the SJA1000 some functions of the Basic CAN behavior have been changed in terms of improved performance. Additionally, many new features have been implemented resulting in interesting functional enhancements, e.g., the receiver functions of the controller.

The SJA1000 is a proper solution for many applications because of its versatility and straightforward architecture which makes it very easy to use. In the following chapters this new approach is presented with several application examples.

2. Stand-alone CAN Controller SJA1000

Based on a modular approach, the SJA1000 can be subdivided into two main groups of functions (Figure 1):

- The CAN Core Block which fully supports the CAN 2.0B protocol standard [2].
- The Message Buffer and Control Block of the device which includes the complete transmit and receive path as well as the interface to the host microcontroller.

The Message Buffer (Transmit Buffer, Receive FIFO and Acceptance Filter) and the Control Block (Interface Management Logic) of the CAN controller allows the support of two modes: A PCA82C200 compatible mode (BasicCAN mode) and a mode that provides many new attractive functions (PeliCAN mode).

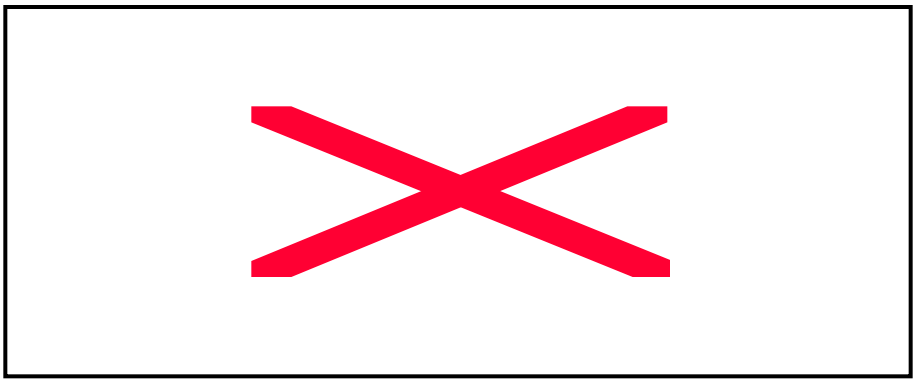


Figure 1: Block Diagram of the SJA1000

2.1. PCA82C200 Compatibility

Achieving compatibility to the PCA82C200 has been very important during the SJA1000 development. Therefore, the new CAN controller provides the default BasicCAN mode following a hardware reset. In this mode all the functions and features of the PCA82C200, e.g., programmable CAN bit-rate (up to 1Mbit/s), flexible microcontroller interface, etc. are available.

Consequently, the SJA1000 is a fully compatible replacement for the PCA82C200. In other words, existing hardware and software can be used without any change. In addition, to the functions necessary to achieve full software compatibility, some features available in PeliCAN mode can now be used in BasicCAN mode as well. These extra features do not influence the compatibility to the PCA82C200. However, they make the device much more flexible even in BasicCAN mode, for explanation see below.

´CAN 2.0B passive´

Extended Frame Messages are now tolerated in BasicCAN mode (´CAN 2.0B passive´). Therefore, modules operating with the PCA82C200 have the chance to work additionally in networks with 29-bit messages. Due to the extended oscillator tolerance the bit timing of the SJA1000 allows the usage of ceramic resonators.

More CAN Performance

Compared to the performance of the PCA82C200 the new 64 byte Receive FIFO of the SJA1000 improves the system behavior significantly. Furthermore the clock frequency range has been extended to allow crystal frequencies up to 24 MHz. This enables an improved bit timing programming which has a positive impact on the maximum bus length. Compared to applications with the PCA82C200 it is now possible to extend the length of bus cables by approximately 40% at 1Mbit/s.

2.2. New Functions of the SJA1000

The new functions in PeliCAN Mode are entered by setting a certain bit in the Clock Divider Register during initialization. From this moment, the register set behaves completely different and the message buffers are now capable of supporting ´CAN2.0B active´. Furthermore, acceptance filtering now comprises every identifier bit, including that of extended frame messages. In addition, new functions useful for versatile applications have been added. The following list gives a survey of the main PeliCAN features:

· CAN 2.0B active	‘ Extends application field to networks with 29-bit identifiers
· Receive FIFO (64 byte)	‘ Lengthens usable time for interrupt services
· Enhanced Acceptance Filter	‘ Reduces unwanted interrupts for the CPU
· Error Analysis Functions	‘ Offers new diagnostic facilities
· Arbitration Lost Capture	‘ Offers new diagnostic facilities
· Single Shot Transmission	‘ Allows fast reloading of transmit buffer
· Listen Only Mode	‘ Allows hot-plugging and monitor functions
· Receive Sync Pulse Generation	‘ Supports system clock synchronization
· CAN Self Tests	‘ Supports functional tests of complete CAN nodes

The next sections briefly explain the PeliCAN mode features.

´CAN 2.0B active´

The SJA1000 supports all frame types specified in the CAN 2.0B standard. In other words, both 11-bit identifier and 29-bit identifier messages can be transmitted and received.

Receive FIFO

In almost every CAN node the host microcontroller has many other tasks besides CAN communication. Some of these tasks may have a higher priority. Therefore, it can be necessary for the application to disable CAN receive interrupt services for a certain time. Consequently, CAN controllers may run into data overrun situations. With the Receive FIFO concept in the SJA1000, the probability of a ´Data Overrun´ condition is reduced extremely. Depending on message format and data length, the 64 byte Receive FIFO is now able to store up to 21 messages without overrun.

Enhanced Acceptance Filter

The receive interrupt rate for a microcontroller is an important parameter that has to be taken into account during system development. With the help of 4 Acceptance Code Registers and 4 Acceptance Mask Registers the acceptance filter of the SJA1000 ensures that the number of unwanted interrupts is decreased significantly.
Two different filter modes are available, allowing either Single or Dual Filter Mode configuration. Figure 2 shows that in Single Filter Mode the SJA1000 uses one filter and in Dual Filter Mode two filters simultaneously for acceptance filtering. Depending on the CAN frame type (Standard or Extended Frame) different bits are relevant for filtering in each mode. Additionally, groups of identifiers can be defined in both modes with the Acceptance Mask Registers.

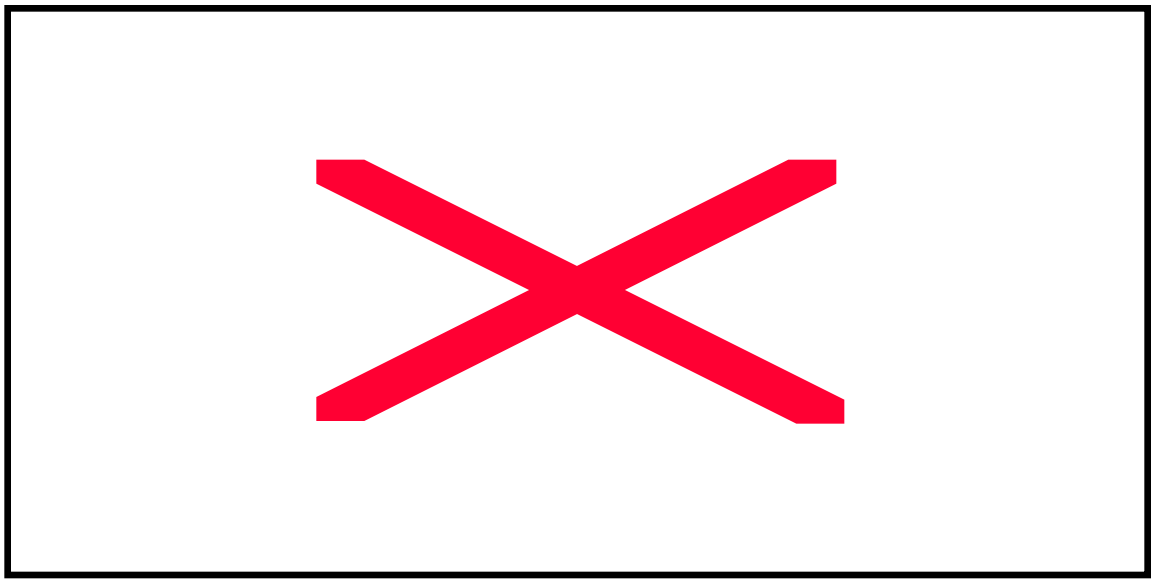


Figure 2: Relevant bits for acceptance filtering

Error Analysis Functions

Whenever a CAN bus error occurs, the corresponding bus error interrupt is activated. Simultaneously, the current bit position and the type of error are captured into the Error Code Capture Register. This register distinguishes four different types of errors: bit-, stuff-, form- and other errors. In addition, the current bit position is stored in this register. This data is fixed until the application software has read the contents. With the help of this register along with the readable error counters the user can get useful information about the performance of the node or system respectively.

Arbitration Lost Capture

Any time the CAN controller has lost the arbitration process, the current bit position is captured into the Arbitration Lost Capture Register. If the Arbitration Lost Interrupt has been enabled, the SJA1000 generates an interrupt for the CPU. With the help of this facility the user is able to monitor each lost CAN bus access in more detail.

Single Shot Transmission

Upon lost arbitration or bus error, the SJA1000 always performs automatic re-transmission of messages. However, with the Single Shot Transmission command the message is transmitted only once even if one of these conditions occurs. Together with Arbitration Lost Capture, Single Shot Transmission is beneficial for situations where re-transmission of messages is not useful.

Listen Only Mode

In Listen Only Mode the CAN node is able to receive messages without giving an acknowledgment. Whenever the SJA1000 enters this mode the status of the error counters is frozen and the CAN controller operates like in error passive mode. Since the controller does not influence the CAN bus in this mode the device is capable of functioning like a monitor or for automatic bit-rate detection,as explained in

Receive Sync Pulse Generation

Upon successful reception of a message a pulse is generated during the last bit in End Of Frame. This pulse can be effectively used for global clock synchronization, as illustrated in chapter 4.

CAN Self Tests

In contrast to other CAN controllers the SJA1000 is capable of receiving its own transmit messages. In Self Reception Mode the device can perform a complete CAN node check with two options:

- Local self test without any other active nodes on the bus (acknowledge is not needed) and
- Global self test in a running system (acknowledge is needed).

In both cases the complete transmit and receive data paths can be checked. This check also includes the acceptance filtering and transceiver functions. The result of a such simple comparison between transmit and receive message could be to validate the correct functionality of the complete node and the correct connection to the bus.

3. Diagnostics & System Optimization

Diagnostics is becoming an increasingly important subject for networks. More and more measures concerning the performance of each single node and the entire system are being required. The SJA1000 provides all the functions necessary to perform a local diagnosis facility in every single CAN node.
CAN controllers compatible to the CAN 2.0B standard have already implemented comprehensive error detection mechanisms. The whole process of error handling is fully automatic, without CPU involvement. Only error warning signals indicate that a certain level of errors has been exceeded. However, this is not enough for performing a detailed analysis.
Beyond what is already achieved by the CAN error confinement, the SJA1000 provides many new functions for error analysis, see chapter 2.2 (Error Analysis Functions) and [3]. Details about a certain error situation and the reason for an error can be effectively used for diagnostic purposes and system optimization.

Arbitration is running as a background process in CAN controllers. In case of arbitration lost, messages are re-transmitted automatically. To avoid frequent re-transmissions, message priorities have to be optimized for a certain system by proper identifier assignments.
With the Arbitration Lost Interrupt and Arbitration Lost Capture Register, the SJA1000 provides the necessary instruments for the analysis of message priorities in a system.

The following sub-chapters describe how system optimization and maintenance purposes are supported with the features in PeliCAN mode.

3.1. System Optimization Approach

The process of system optimization is performed mainly during the prototype phase but it can also be used during normal system operation. In both cases, the performance of communication is analyzed and can be improved step by step with corrective actions. For the SJA1000 special attention has been paid for supporting this approach. Via several registers the user can have a deep insight into the error confinement of the CAN controller. Advanced features of the PeliCAN mode such as; Error Code Capture Register, Readable Error Counters and Error Interrupt let the user obtain details of a certain error condition on the CAN bus and the reason for it.

With the help of the described features it is possible to verify the performance of single nodes and the requirements of the entire system. Additionally, together with bus failure management on the physical layer level [4], important information about the system performance can be provided. Typical examples include:

- System reaction time
- Frequency of errors
- Distribution of errors
- Number of re-transmissions due to arbitration lost

Subsequently, the performance of the system is optimized with corrective actions.

3.2. System Maintenance

Generated diagnostic protocols can be read during maintenance, whereby detailed reports from the SJA1000 may be translated directly into a description which allows the determination of the error and its location (Figure 3). The translation is typically done with a pre-defined table within the diagnostics software. This approach speeds-up the process of discovering and localization error sources in a network.

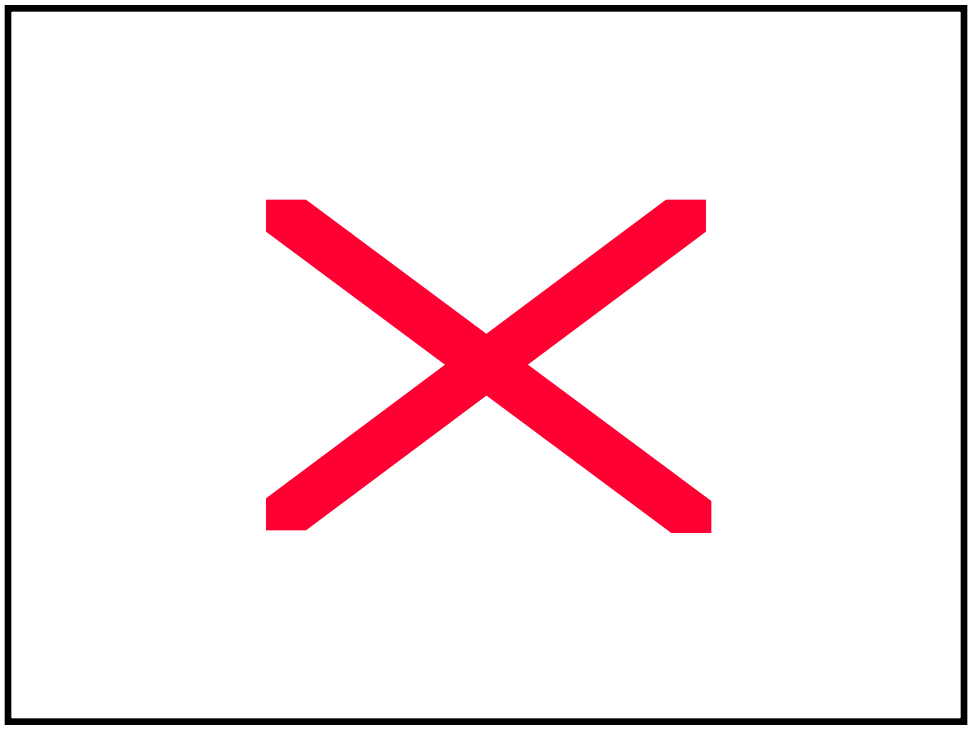
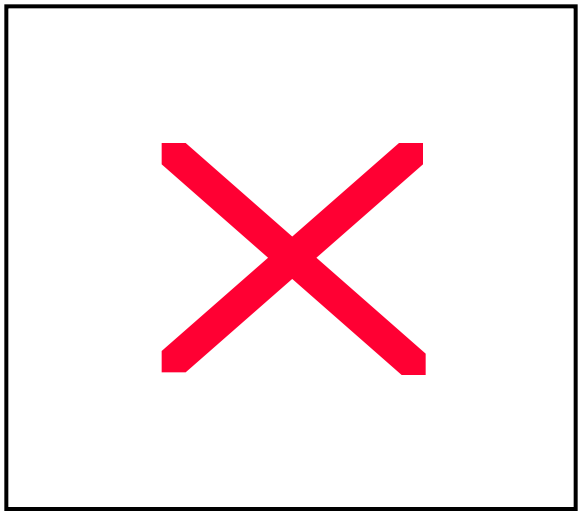


Figure 3: Examples of Error Codes and possible error sources

4. Global Clock Synchronization

As already described in chapter 2.2 (CAN Self Tests), the self reception feature of the SJA1000 can be used for a complete functional CAN node check. In addition, this feature is also well suited for clock synchronization of entire systems, which is shown in this chapter.



In distributed systems it is difficult to implement a system wide clock without having an extra synchronization line [5]. All nodes connected to the bus have local clocks with clock drifts. In this example (Figure 4) one node in the network is assumed to operate as a ´master´ clock. The remaining clocks in the network are synchronized to the value of the master clock.
Both the Self Reception Request feature and the fact that each SJA1000 is able to generate a pulse at a definite time upon message reception can be used to support clock synchronization in distributed systems.
In the following section a simplified procedure for global clock synchronization is presented.

Figure 4: System with Global Clock Synchronization

In Figure 5 a system master transmits a ´Self Reception Message´ onto the CAN bus. After message reception, each node, including the master, generates a Receive Sync Pulse. In every slave node it is used, i.e., to reset the timers. Simultaneously the master node uses this pulse to capture the master clock value t_M.
In a next step this value is sent as a ´Reference Time Message´ to all slaves. A simple adder function in every slave, followed by reloading all timers with t_S synchronizes the system wide clock.

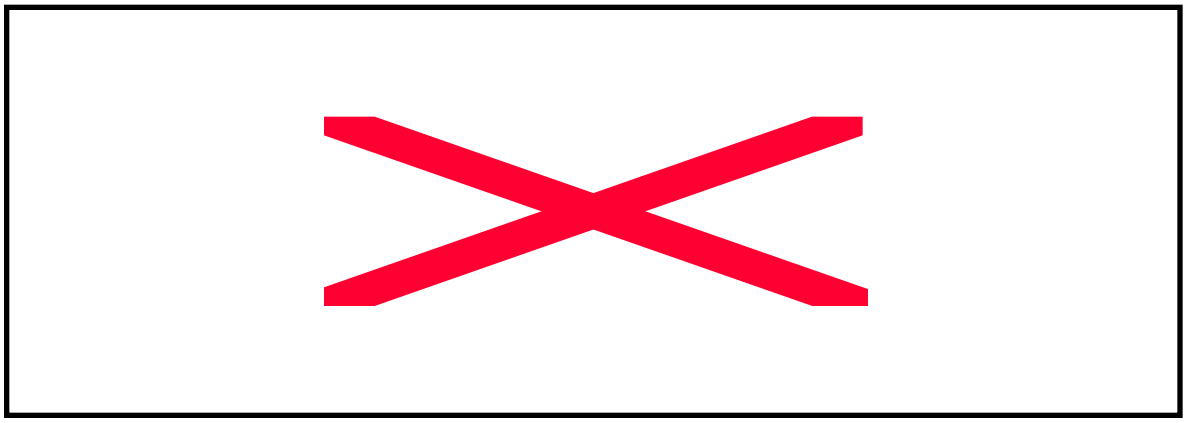


Figure 5: Timing Diagram during System Synchronization

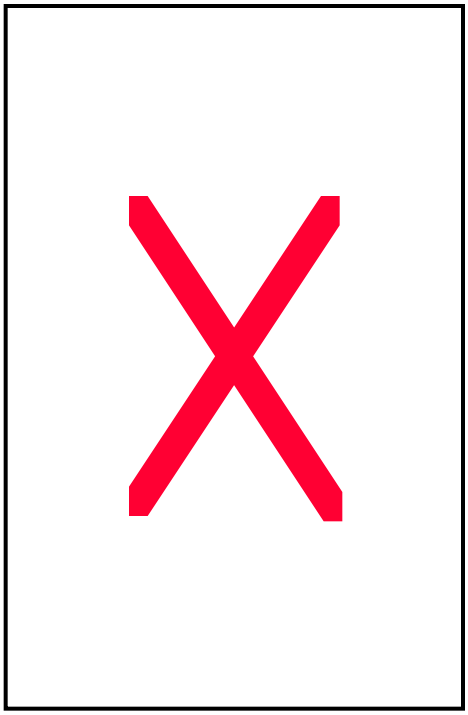
The major advantage of this concept is the simplicity of implementation without complicated time stamp handling. No software cycle count is necessary because critical paths are hardware controlled and therefore deterministic. Furthermore it is independent of network parameters. Interrupt events may happen during the complete period without influencing the synchronization process.

5. ´Plug & Play´

`Plug & Play` solutions are becoming more and more important. A bus node which is plugged onto an unknown network first has to establish a correct bit-rate. Therefore an automatic bit-rate detection is needed.

Automatic Bit-Rate Detection

The major drawback of existing trial and error concepts for automatic bit-rate detection is the generation of CAN error frames which is not acceptable. The SJA1000 supports these requirements with new features of the PeliCAN mode. This section briefly describes an example for automatic bit-rate detection without influencing running operations on the network.
In Listen Only Mode, the SJA1000 is neither able to transmit messages nor to generate error frames. Only message reception is feasible in this mode. A pre-defined table within the software contains all possible bit-rates including their bit-timing parameters. Before starting message reception with the highest possible bit-rate, the SJA1000 enables both receive and error interrupts (Figure 6).
In case of one or more errors on the CAN bus, the software switches to the next lower bit-rate.
Upon successful reception of a message, the SJA1000 has detected the right bit-rate and can switch to normal operating mode. From now on this node is able to operate as any other active CAN node in the system.



6. Summary

With the SJA1000, Philips Semiconductors continues the development of stand-alone CAN controllers. Applications currently running with the PCA82C200 stand-alone CAN controller can change directly to the SJA1000. By default it is compatible to the PCA82C200 and neither hardware nor software changes are required. Furthermore, new applications are possible with new attractive features, especially when System Optimization, Diagnosis and Maintenance purposes are important. Additionally, functions supporting self tests, global clock synchronization and automatic bit-rate detection have been added to enhance functionality.

References

[1] Data Sheet, PCA82C200 Stand-alone CAN Controller, Philips Semiconductors, October 1990.
[2] CAN Bus Specification 2.0, Philips Semiconductors, 1991.
[3] Data Sheet, SJA1000 Stand-alone CAN Controller, Philips Semiconductors, April 1997.
[4] Mores, R: Advanced Bus Failure Management for Physical Layer of CAN, SAE Technical Paper 960119, Detroit, USA, 1996.
[5] Gergeleit, M; Streich, H: Implementing a Distributed High-Resolution Real-Time Clock using the CAN-Bus, International CAN Conference, Mainz, Germany, September 1994.