Higher flexibility in automotive networks through CAN FD

Radoslaw Watroba, Rolf Nandlinger

Authors
Radoslaw Watroba
Rolf Nandlinger

STMicroelectronics
Application GmbH
Bahnhofstr. 18
DE-85609 Aschheim-Dornach
Tel.: +49-89-46006-0

Link
www.st.com

Introduction

The continuous increase of connectivity in automotive networks forces higher data rates. One of the possible solutions to resolve this dilemma is the CAN FD protocol (CAN with flexible data rate). First CAN FD protocol controller implementations integrated in automotive 32-bit micro-controllers are under validation. The semiconductor manufacturers are confronted with new timing requirements to be considered during the system-on-chip integration, which collide with the required current consumption limits. Currently, power train as well as chassis and safety applications are targeted by CAN FD. The carmakers are longing for first implementations for evaluation. The ECU suppliers focus on the reduction of the end-of-line programming times.

The concept behind CAN FD is a transmission of one CAN frame at two speeds. Whereas the arbitration field is transferred at maximum 1 Mbit/s the following payload is accelerated to higher bit-rates e.g. 4 Mbit/s, which is targeted for the first use cases. In contrast to the CAN protocol as defined in the ISO 11898-1 the maximum payload length according to CAN FD has been extended from 8 bytes to 64 bytes. Delays caused by the mandatory network idle times and delays due to the arbitration process are avoided increasing the average data rate as well.

1 Mbit/s is often mentioned as the upper limit for CAN networks. However, the experienced and practiced bit-rate in automotive environments is 500 kbit/s. A robust arbitration and data transmission of the non-deterministic communication is linked to constraints, which have to assure the worst acceptable electrical signal integrity. In addition, automotive networks reflect years of deep experience, therefore, in the most cases, a migration to a CAN FD network in terms of using higher bit-rates should not be assumed. However, customers lacking bandwidth and not willing to go above 500 kbit/s consider the 64-byte payload as an attractive feature of CAN FD.

The CAN FD protocol was also developed to close the gap between CAN (max. 1 Mbit/s) and Flexray (max. 10 Mbit/s). The latter needs dedicated transceivers. Until the new generation of CAN transceivers supporting bit-rates higher than 1 Mbit/s will be available, the use of actual transceivers in mass production at lower data rates is obligatory. Concerning programming with higher switching frequency, the semiconductor manufacturers were requested to modify their actual transceiver specifications and increase the upper bit-rate limit for this use case. The CAN FD introduction requires at least the use of a CAN-FD-capable protocol controller, in fact in each CAN node. Otherwise the communication would break down due to CRC (cyclic redundancy check) errors interpreted by CAN controllers according to ISO 11898-1 during the CAN FD mode. In April 2012 Robert Bosch has released the CAN FD protocol specification v. 1.0. The goal is to transfer that protocol specification to an ISO standard. Since some open questions still exist, for instance, regarding the physical layer parameters or conformance testing, working groups with individual focus were initiated by the CAN in Automation (CiA) organization.

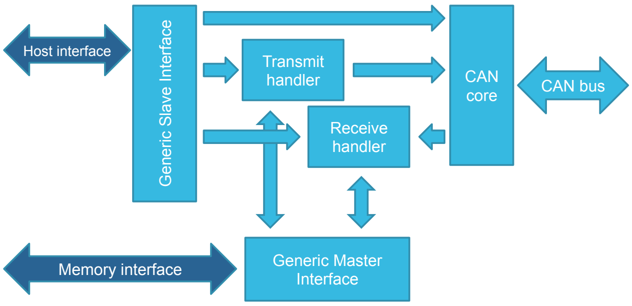

Figure 1: M_CAN module

First CAN FD protocol controllers

The decision to implement a CAN FD protocol controller was made very early. ▷

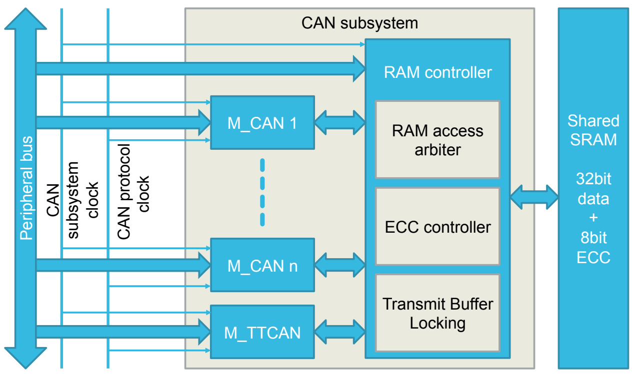

Figure 2: SPC57 CAN sub-system

CONTROLS

Ethernet Controller EC1000

Compact PLC controller with CODESYS 2.3 or 3

Our EC1000: A strong CODESYS controller for the modular CANtrol EC control system. The high performance EtherCAT I/O modules are lined up directly next to the PLC controller. The EC1000 offers state-of-the-art industrial Ethernet technology combined with a **CAN/CANopen** interface. It is a control platform which combines tried and tested CAN components with state-of-the-art EtherCAT modules to create an optimal control solution.

→ Optimal performance with EtherCAT up to the I/O terminal
→ Easy visualisation with CODESYS Visu and Berghof Ethernet terminals
→ Fast connections with Ethernet/EtherCAT®, CANopen and serial
→ EC1000 starter package with CODESYS V3. Information at: www.berghof.com/starterpaket

We are exhibiting at:
Hall 9 | Stand D62

Berghof Automationstechnik GmbH
Harretstrasse 1
72800 Eningen | Germany
www.berghof.com

Currently, STMicroelectronics is developing the SPC57 32-bit micro-controller family based on the 55-nm and 40-nm in-house embedded flash technology for power train as well as chassis and safety applications. Bosch's M_CAN IP was chosen. Internal validation of the first silicon and therefore of the CAN FD protocol on a target hardware supporting 8-byte payload has been started. Meanwhile the M_CAN IP has been refined and the maximum payload length was extended to 64 bytes. Consequently all developed micro-controllers will switch to the most recent M_CAN IP with 64 byte payload support.

The MCU family implements several M_CAN instances depending on the micro-controller derivative. In short, each M_CAN instance consists of a CAN core, transmit and receive handler, generic master and slave interface and additional control and synchronization logic. While the generic master interface connects the M_CAN to the external 32-bit message RAM, the connection to the host CPU is done by the generic slave interface.

MCU's CAN sub-system

The CAN sub-system of the SPC57 devices includes several M_CAN instances and a RAM controller as interface to the shared CAN RAM, which includes the transmit and receive buffers as well as the configurable CAN filter elements of each M_CAN instance. The RAM controller consists of the SRAM interface, an active transmit message buffer protection, an ECC controller and additional logic, which handles the arbitration of the RAM access requests by the M_CAN instances and the CPU.

Each M_CAN generic slave interface and the CAN RAM controller slave interface is connected to an individual slot on the 32-bit peripheral bridge while the M_CAN generic master interfaces are connected to the CAN RAM controller. The shared CAN RAM array is organized by a 32-bit data and an 8-bit ECC. Each M_CAN instance supports a maximum RAM-size of 1216 x 32-bit words (for M_CAN including TTCAN 1344 x 32-bit words). The message RAM is used by the M_CAN instance for storing of the 11-bit and 29-bit CAN message filters, two receive FIFOs, one receive buffer, one transmit event FIFO and one transmit buffer. In case the time-triggered CAN communication feature of the M_CAN IP is used, additional trigger RAM memory must be configured.

Each section of the message RAM is fully configurable in terms of the start address and the size. There is one exception regarding the transmit buffers. Those special sections of the CAN RAM are reserved and cannot be configured. Each element of those buffers can be protected by an M_CAN hardware signal against CPU write access until the last valid data was transmitted. The maximum size supported by the M_CAN is shown in Figure 3. The overall size of the message RAM module is assigned during the micro-controller derivative integration depending on the targeted application requirements. The flexible message RAM handling allows an optimized chip-size required by the CAN RAM, while the usage of this memory can be configured by the user according to the application requirements.

CAN RAM bandwidth

For systems with multiple M_CAN instances connected to one shared CAN RAM a consideration of the overall bandwidth requirements for the memory interface is mandatory. Any bottleneck in the data transmission has to be avoided in case all M_CAN instances are running at high speeds and high message loads. On the other hand, oversizing of the system in terms of increasing the operating frequency should be avoided to limit the overall current consumption. An M_CAN-specific theoretical worst-case data traffic to the shared CAN RAM is given by the following conditions:

- CAN FD mode
- 1 Mbit/s during arbitration
- 4 Mbit/s during payload transmission

	Configurable number of elements	32bit words per element		
11bit Message Filter	0…128	1	M_CAN up to 1216 x 32bit words	M_TTCAN up to 1344 x 32bit words
29bit Message Filter	0…64	2		
Rx FIFO0	0…64	4		
Rx FIFO1	0…64	4		
Rx Buffers	0…64	4		
Tx Event FIFO	0…32	2		
Tx Buffers	0…32	4		
TT-CAN Trigger Memory	0…64	2		

Figure 3: Supported shared memory size by M_CAN

Start Address Offset (byte)	End Address Offset (byte)	Size (Elements)	CAN Block/Sub Block	
SPC574K72				
0000_{HEX}	$0CFF_{HEX}$	configurable by software	M_CAN 1	Standard Message Filters
				Extended Message Filters
				Rx FIFO0
				Rx FIFO1
				Rx Buffers
				Tx Event FIFO
$0D00_{HEX}$	$0EFF_{HEX}$	32 (protected)		TxBuffers
$0F00_{HEX}$	$1BFF_{HEX}$	configurable by software	M_CAN 2	Standard Message Filters
				Extended Message Filters
				Rx FIFO0
				Rx FIFO1
				Rx Buffers
				Tx Event FIFO
$1C00_{HEX}$	$1DFF_{HEX}$	32 (protected)		TxBuffers
$1E00_{HEX}$	$2AFF_{HEX}$	configurable by software	M_TTCAN	Standard Message Filters
				Extended Message Filters
				Rx FIFO0
				Rx FIFO1
				Rx Buffers
				Tx Event FIFO
$2B00_{HEX}$	$2CFF_{HEX}$	32 (protected)		TxBuffers
$2D00_{HEX}$	$2EFF_{HEX}$	32 (protected)		M_TTCAN triggers

Figure 4: Example of the shared CAN RAM as implemented in an SPC57 derivative

Figure 5: SPC5x micro-controller

- 128 active filter elements per CAN for an 11-bit CAN-ID
- 32 transmit buffers per M_CAN
- DLC = 0
- 50-% CPU access time to the shared RAM

Assuming the mentioned conditions being applied to all M_CAN implemented instances an internal RAM bandwidth of 15 Mbit/s for each module would be mandatory. A typical CAN subsystem consisting of three or four M_CAN modules would require a clock frequency higher than accepted in terms of current consumption aspects. Based on the targeted powertrain applications the worst-case scenario can be changed by making the following assumptions:

- Maximum two M_CAN instances running in CAN FD mode (1 Mbit/s arbitration and 4 Mbit/s payload)
- Up to three M_CAN instances running in standard CAN mode (500 kbit/s or 1 Mbit/s)
- DLC ≥ 1

In this scenario the bandwidth requirements for the shared CAN RAM become less demanding and the CAN subsystem can be operated at 50 MHz, which is an appropriate frequency for automotive applications.

CAN FD activities in automotive industry

The internal validation at STMicroelectronics of an 8-byte payload CAN FD controller has been started. The availability of the first silicon implementing CAN FD is an important milestone to prove the CAN FD protocol controller on the target hardware.

The carmakers are very open-minded towards CAN FD and consider their internal evaluations as soon as the engineering samples will be available. Some of the carmakers go even beyond that and, already today, plans for the next generation networks supporting CAN FD are made.

The ECU suppliers currently focus on the reduction of the end-of-line programming time. Implementation of CAN FD as a faster backbone network has a lower priority. The semiconductor manufacturer is sustainable committed to automotive and will successively expand the SPC57 micro-controller family including CAN FD protocol controllers. Currently powertrain applications are targeted by the first developments. In parallel, micro-controller derivatives fulfilling the chassis and safety requirements are under development. The internal validation of a micro-controller derivative with a 64-byte-payload CAN FD controller is planned for Q4/2013. Slightly later delivery of engineering samples to strategic partners is planned.