

CAST

CAN-CTRL

Flexible CAN 2.0 Bus Controller Core

Features

CAN Specification 2.0B

- Standard and Extended Data and Remote Frames

Enhanced Functionality

- Error Analysis features enabling diagnostics, system maintenance and system optimization
 - Last error type capture
 - Arbitration lost position capture
 - Programmable Error Warning Limit
- Listen-Only Mode enables CAN bus traffic analysis and automatic bit-rate detection
- Optionally extendable to a Dual CAN controller for applications with two CAN interfaces

Flexible Message Buffering and Filtering

- Configurable number of receive buffers, 2 to 31
- One high-priority transmit buffer
- Configurable number of lower-priority transmit buffers, 0 to 16
- Configurable number of independently programmable 29-bit acceptance filters, 1 to 16

Easy to Use and Integrate

- Programmable data rate up to 1 Mbit/s
- Programmable baud rate prescaler. 1/2 up to 1/256
- Single Shot Transmission Mode for lower software overhead and fast reloading of transmit buffer
- Flexible programmable interrupt sources
- Generic 8-bit host-controller interface and optional AMBA-APB
- Buffers can be implemented as Flip-Flops or RAM

Zero Risk

- Verified with the Bosch reference model
- Link to commercial bus drivers (e.g. PCA82C250T by Philips)
- Multiple times production proven

Efficient and Portable Design

- Available in RTL, and portable to ASIC and FPGA technologies
- Size of approximately 12,000 gates.

Implements a CAN protocol bus controller that performs serial communication according to the CAN 2.0A and 2.0B specifications.

The CAN protocol uses a multi-master bus configuration for the transfer of frames between nodes of the network and manages error handling with no burden on the host processor. The core enables the user to set up economic and reliable links between various components. It appears as a memory-mapped I/O device to the host processor, which accesses the CAN core to control the transmission or reception of frames.

The CAN core is easy to use and integrate, featuring programmable interrupts, data and baud rates; a configurable number of independently programmable acceptance filters; and a generic processor interface or optionally an AMBA-APB interface. It implements a flexible buffering scheme, allowing fine-tuning of the core size to the requirements of each specific application. The number of receive buffers is synthesis-time configurable from 2 to 31. Two types of transmit buffers are implemented: a high-priority primary transmit buffer (PTB) and a lower-priority secondary transmit buffer (STB). The PTB can store one message, while the number of included buffer slots for the STB is synthesis-time configurable 0 to 16 slots.

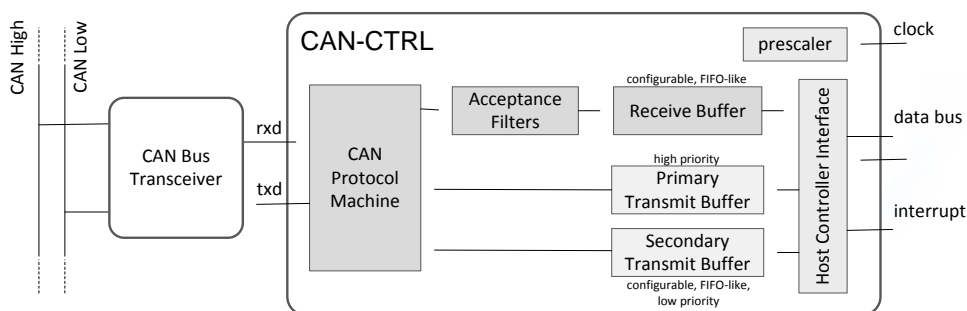
The core implements functionality similar to the Philips SJA1000 working with its Pelican mode extensions, providing error analysis, diagnosis, system maintenance and optimization features.

The CAN core is robustly verified and proven in multiple production designs.

Applications

The CAN protocol was designed specifically for automotive applications but is now also used in other areas, such as railways, industrial automation, and medical equipment.

Block Diagram



Functional Description

The CAN bus core is founded on the basic CAN principle and meets all constraints of the CAN-specification 2.0B.

Several 13-byte buffers are used for buffering received or transmitted messages. The number of buffers can be selected before synthesis. Selecting a large number of buffers disables the need for real-time reaction to CAN messages for the host processor, which significantly eases software development of the system application.

The included high-priority transmit buffer can be used to transmit an important message as fast as possible, even if several lower-priority messages are pending for transmission.

The host interface contains all necessary registers for controlling and configuring the core. The host is able to read and write all registers as a conventional RAM in memory mapped mode.

The interface to the host is software configurable. All events on the CAN data bus or in the CAN controller core are signaled using an interrupt. Every interrupt source may be individually enabled or disabled. The CAN controller core contains three software-programmable 29-bit acceptance filters that can be used to block unwanted CAN messages, which reduces the load to the host controller.

The host controller interface is connected with the memory module by an 8-bit data and a 6-bit address bus. This enables easy interfacing to many host controller types, and therefore, quick integration with a microcontroller.

Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Verification

The core has been certified by a Bosch reference model and has been production proven multiple times.

Implementation Results

CAN-CTRL core reference designs have been evaluated in a variety of technologies. The following are sample results optimized for area with three message buffers and three acceptance filters implemented as flip-flops.

Technology	Approx. Area	Frequency
TSMC 130nm process	11,930 gates	214 MHz
TSMC 90nm process	10,670 gates	400 MHz
TSMC 65nm process	12,840 gates	> 500 MHz

Deliverables

The core includes everything required for successful implementation:

- VHDL or Verilog RTL source code
- Post-synthesis EDIF (netlist licenses)
- Testbenches
 - Behavioral tests
 - Bosch reference model
 - Post-synthesis verification
- Simulation scripts
- Synthesis scripts
- Documentation