

CAN Bus Features

Item	Feature	Benefit
Ethernet-Interface	Full Duplex Gigabit Ethernet. Automatic speed negotiation and downshift for 100Base-TX, and 1000Base-T data rates. Automatic detect and MDI Crossover	Low cost, available, high speed, networked data bus 1000 times faster than CAN and supported by almost all computers.
Communication Protocols	Ethernet with MAC flow control protocol. UDP/IP, ARP and ICMP.	UDP is a fast low latency protocol and part of the TCP/IP family, supported by almost all operating systems.
Multi-user Interface	Up to 10 separate applications can logon to the module and access and control the resources.	System design can be simplified by dividing it into separate applications, which can be developed independently and run on the same or separate computers
System Control and Status Management	System tasks performed in parallel by dedicated logic implemented in a Field Programmable Gate Array (FPGA).	Avoids processing bottlenecks, which can occur with the sequential processing of microprocessors.
CAN Controller	NXP SJA 1000	Highly popular, well proven silicon
CAN Channels	4	Sufficient for most needs.
Identifier	11/29 bit	Support for CAN-2A and CAN-2B Modes
Timestamp	40 bit counter with 1 μ s resolution under hardware control. Each message is time-stamped on the leading edge of each message	Ample resolution 1 Mb/s CAN bus and the counter will not rollover for 12 days.
Queued Transmissions	Asynchronous FIFO transmit facility with storage for more than 42 messages	Offloads processing effort: Transfer and Forget, sequential message queue.
Scheduled Transmissions	Individually configured periodic message scheduler with a repetition rate programmable from 1 ms to 16 seconds	Offloads processing effort: Configure once and update message data, only when necessary.
Reception Response Transmissions	Individually configured to send messages in response to specific receive Identity. Receive messages can be optionally filtered by ID (11 or 29 bit) and RTR.	Offloads processing effort: Configure and forget, or occasionally update data.
High Priority Transmissions	Asynchronous transmission facility where messages are sent immediately avoiding the Transmit Queue, except where arbitration dictates that it should wait for a more dominant message to be sent.	Allows high priority messages to be transmitted without delay The Queued Transmission port can then be filled with many messages without concern for the delay it may cause, when a high priority message must be sent.
Transmit Arbitration	Arbitration between different transmission sources gives priority to the most dominant Identity.	Allows messages to be sent in the order of priority.
Message Data Transfer Arbitration	Data transfer to the Transmit Message Buffer is arbitrated.	Ensures Message Fidelity. Arbitration avoids message corruption.
Data Transfer Latency	From Application Software to CAN operation: 50 μ s (Typical) From CAN operation to Application response: 80 μ s (Typical) All measurements include the Host Test Computer response time of 40 μ s. Note that all high priority tasks are performed in hardware and have very little latency.	Low latency is an essential requirement for many applications, particularly those involving feed back control. The latency in transferring data to and from the module is in the order of the time it takes the CAN bus to transfer a message when operating at 1 Mb/s.
Message Monitoring and Reception	Status information can be captured for all interrupt events which include: Reception, Transmission and Bus Errors. Additional status information is captured when operating in CAN-2B mode. Message Data is captured for all Receive messages. It is also possible to capture Transmit Data.	Offloads processing effort: Status and Data are transferred automatically Cyclic Buffers without any direct processor support.
Application Status and Data access	Status and Data are automatically transferred to Cyclic Buffers, which can be automatically transferred to the host application at regular intervals.	Offloads processing effort: The processes of capturing and transferring information to host applications is highly automated.