# BRIGOSHA

**EMP. NAME: - NARESH GANGARABOINA**
**EMP. ID: - 20220324**

**EMP. NAME: - NIRANJAN DHAGE**
**EMP. ID: - 20220318**

## CAN HARDWARE

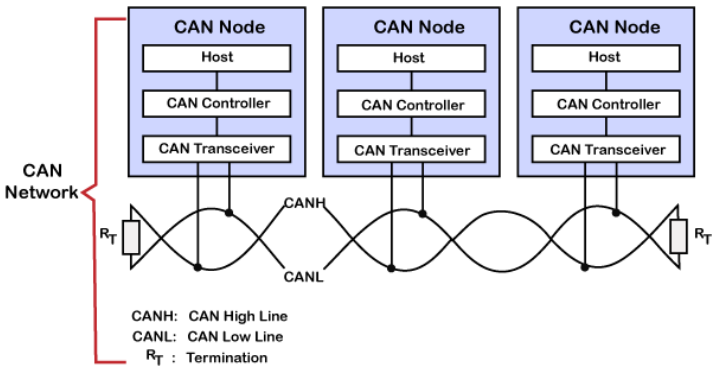
## INDEX

- **INTRODUCTION**

Controller area network is an electronic communication bus defined by the ISO 11898 standards. Those standards define how communication happens, how wiring is configured and how messages are constructed, among other things. Collectively, this system is referred to as a CAN bus.

Traditionally the CAN bus was primarily used in automotive applications. Today, CAN is one of the best choices for "embedded networking applications" that need to communicate between several embedded 8-bit and 16-bit microcontrollers. Biggest current growth sectors are "embedded machine control" applications: home appliances, industrial machines or other machinery that use multiple microcontrollers that need to communicate with each other. One of the latest examples in this arena is a high-end coffee machine that uses microcontrollers that are interconnected via CANopen.

- **WHAT IS CAN HARDWARE**




The CAN protocol is a standard designed to allow the microcontroller and other devices to communicate with each other without any host computer. The feature that makes the CAN protocol unique among other communication protocols is the broadcast type of bus.

- **CAN REQUIRMENTS**

As we know CAN is TWO wire twisted pair differential bus, one is CAN high and another is CAN low. It means this is important requirement.

Following are some requirements:

Host/MCU

CAN Controller

CAN Transceiver

- **CAN KEY ADVANTAGE**

- **Built-in Error Detection**

One of the key features of the CAN bus protocol is that it supports centralized control over electronic devices that are connected to the network. In the CAN bus physical layer, each electronic device is called a node. Nodes can communicate with other nodes on the network, and each node requires a microcontroller, CAN controller, and CAN transmitter.

A big advantage of the CAN bus compared to other network solutions is the price/performance ratio. Price wise, CAN is the most affordable network next to a regular serial channel. As a rule over thumb one can say that it costs about $3 to CAN-enable an existing microcontroller design. Replacing an 8-bit or 16-bit microcontroller with one that features a CAN interface costs about $1. An additional $1 is needed for the transceiver (line driver for twisted pair) and another $1 for connectors and additional PCB area.

- **MAXIMUM CABLE LENGTH**

At a speed of 1 Mbit/s, a maximum cable length of about 40 meters (130 ft.) can be used. This is because the arbitration scheme requires that the wave front of the signal be able to propagate to the most remote node and back again before the bit is sampled. In other words, the cable length is restricted by the speed of light. A proposal to increase the speed of light has been considered but was turned down because of its inter-galactic consequences.

Other maximum cable lengths are (these values are approximate):

- 100 meters (330 ft) at 500 kbit/s
- 200 meters (650 ft) at 250 kbit/s
- 500 meters (1600 ft) at 125 kbit/s
- 6 kilometers (20000 ft) at 10 kbit/s

If optocouplers are used to provide galvanic isolation, the maximum bus length is decreased accordingly. Hint: use fast optocouplers, and look at the delay through the device, not at the specified maximum bit rate.

- **HARDWARE FILTERING WITH MATCH AND /OR MASK**

The functionality of hardware filters is very similar on many CAN devices. While receiving a CAN message, the identifier (and sometimes even the data) can be compared to a configured filter. Only if the incoming message matches the filter does the message get stored into a receive buffer. The major differences in filters are usually the width of the filter and if it is a match only filter or also allows a mask to be used.

The filter width specifies how many bits of an incoming CAN message can be processed. For a standard CAN message identifier at least 11-bits are required. For an extended CAN message identifier its 29-bits.

Where a match filter only allows you to do one exact match (for example exactly one identifier), a combination of match and mask allows for filtering on message groups (for example identifiers 0x100 to 0x11F). Usually a bit set in the mask register means that the corresponding bit in the CAN message is a "don't care" value for the acceptance filtering. If a bit is cleared, it MUST match the value in the match register.

- **SUMMARY AND CONCLUSION**

Many CAN controllers offer additional features such as extended error reporting and diagnostic functions or auto-baud detection. For the scope of this article a complete feature comparison is impossible - so we concentrated on the number one feature essential to many CAN applications: receiving messages.

Before choosing a CAN interface for an application do a worst-case analysis. What is the fastest baud rate the node needs to support? How much of the network traffic does need to be worked on? How much additional network traffic is there and can it be completely eliminated by hardware filters? Which percentage of the MCU performance is needed for the CAN communication and which percentage required by the real application running on that MCU.