CAN - How it works

October 16, 2023

Introduction

CAN or rather Controller Area Network is a networking protocol that allows devices to communicate without a master. Every node in the network can send information by specifying the receiver's ID along with the data. The CANBUS is a serial bus with two wires known as CAN LOW and CAN HIGH. Serial means it sends information one bit at a time organized in data frames. In order for two nodes in the network to talk to each other.

Differential Signal

As previously mentioned the CANBUS consists of two wires. The reason for this is to remove noise. If only one cable was used for communicating, noise from vibrations, magnetic fields, etc. can vary the signal's voltage enough to change the bit from 0 to 1 or from 1 to 0. To counteract this issue, CANBUS uses two wires, CAN LOW and CAN HIGH. Instead of using the voltage of a single wire to represent a bit, it uses the voltage difference between the CAN HIGH and the CAN LOW. As the wires should always be routed and twisted together, they will experience the same noise, thus keeping the voltage difference stable. A visual representation looks like this:

The bus usually requires 5V to operate. To represent bits the CAN LOW is

Figure 1: Noise affects both wires equally

then set at either 1.25V or 2.5V, while the CAN HIGH is set to 2.5V or 3.75V. A significant voltage difference (usually 2.5V) represents a 0-bit while no voltage difference represents a 1-bit.

Asynchronous

The CANBUS is asynchronous, meaning each node does not have a clock pin. Instead of sharing a clock, the network must agree on a common baud rate. This allows all devices to interpret the signals correctly as they know the pulse length of each bit sent. If the network's baud rate is decided to be 1 Mbit/s, and then a new node is set up with a baud rate of 2 Mbit/s, it will essentially read double the bits sent or send twice as much as the other nodes can read.

Data Frames

Every message that a node wants to send, has to be formatted using a specific data frame. For CANBUS, the data frames can be divided into the following 8 blocks.

SOF (1 bit)

After every frame, the bus idles for some amount of time. The SOF bit marks the start of a new message and is used to synchronize all the nodes.

Arbitration (12 bits)

The arbitration block consists of an 11-bit identifier and an RTR bit. The first 11 bits specify the ID of the node that should receive or send the data. The lower the value of the identifier is, the higher the priority is.

The last bit is known as the Remote Transmission Request bit. It specifies whether data is being sent or is requested. The bit is dominant (0) when a node requests data. If the bit is dominant, any nodes in the network will receive the data and are free to use it.

Control Field (6 bits)

The control field can be split into three sections. The first bit is the IDE which specifies if the ID is 11 or 29 bits. The standard is to use 11 bits for the ID, but it is possible to decide on a CAN that uses 29-bit IDs. A dominant IDE bit is used to specify 29-bit IDs. The second bit is reserved for future use and is currently not in use. The last 4 bits specify how many bytes are sent in the Data Field.

Data Field (0-8 bytes)

This is where the actual data to be sent or received is specified. The maximum length is 8 bytes (64 bits).

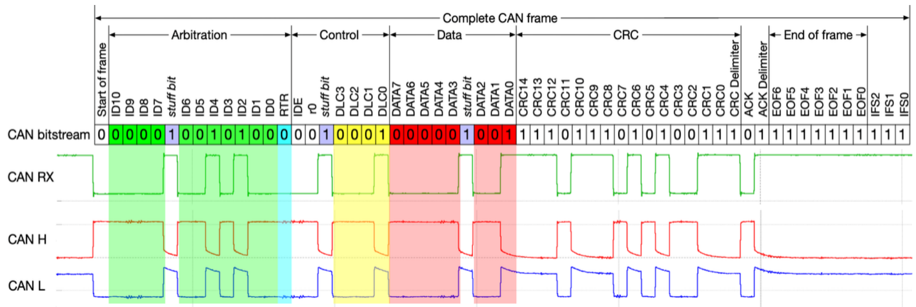

Figure 2: 11-bit identifier data frame.

CRC Field (16 bits)

CRC or rather the Cyclic Redundancy Check is an error detection technique where the first 15 bits are calculated using a polynomial and should represent the data that was sent. The last bit is a delimiter that serves the purpose of helping with synchronization but also to confirm that the frame has been sent correctly.

ACK Field (2 bits)

The acknowledgment field is where the nodes use the received data and the CRC and compare them. The first bit is the acknowledgment bit and the second is a delimiter bit with the same purpose as the CRC delimiter. If the comparison succeeds, every device connected to the network will respond with a dominant bit. If however, any node fails the comparison and therefore does not respond with a dominant bit leaving it recessive, all the receiving nodes will discard the message and the transmitting node will resend the whole frame.

EOF (7 bits)

The EOF bits are used to mark the end of a frame and to synchronize the nodes. During normal operation, when 5 bits of the same logic level are sent consecutively, an opposite bit is "stuffed" into the data frame in order to keep the nodes synchronized. When the EOF is reached, bit stuffing should be disabled. Therefore, if all 7 bits are not of the same logic level, it indicates a stuffing error.

IFS (3 bits)

In the Interframe Space, 3 recessive bits are sent before a new frame can start.

Physical Wiring

The wiring of a CAN BUS is very simple. It consists of a CAN Transceiver and all the nodes. The nodes are connected via CAN HIGH and CAN LOW

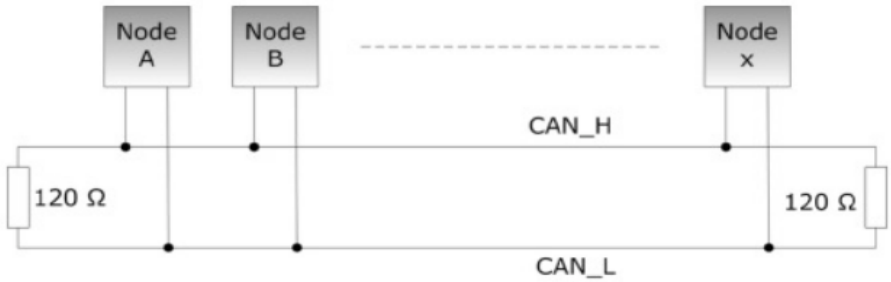

Figure 3: CAN BUS wiring.

and obviously powered in some way. A crucial element in the wiring are the terminating resistors. In both ends of the CAN BUS, there must be a 120Ω resistor in order to stabilize the voltage levels and external noise. Some nodes have internal 120Ω resistors that function as terminating resistors. If that is the case these should be connected on the ends of the bus.

CANOpen

CANOpen is a higher order platform based on CAN...

DeviceNet

DeviceNet is a higher order platform based on CAN...