CAN-bus

The CAN bus is a serial bus which was originally designed for the automotive industry. For the principles of the CAN-bus we refer to the relevant handbooks, for example from the CAN in Automation international users' and manufacturers' group for CAN (Controller Area Network).

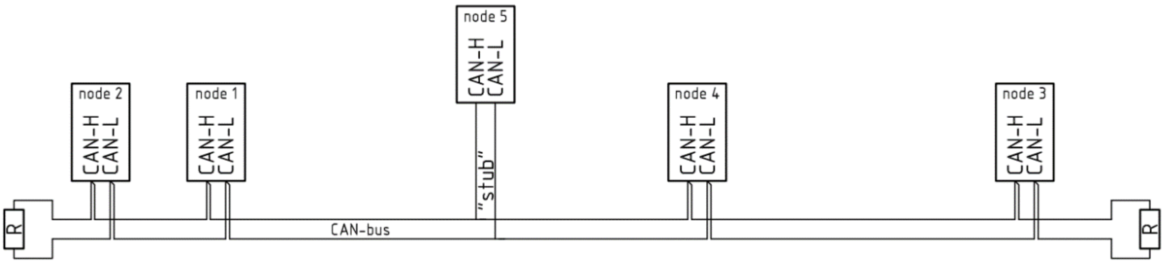

Figure 1: line structure of the CAN bus with termination resistors on both ends

Introduction

Basically, a CAN-bus is a 2-wire connection for exchanging messages between CAN-devices ('nodes'). Data is sent as logical '0' and '1' in a CAN-message. Logical '1' is when the voltage level of the CAN-H data line is higher than the voltage level on the CAN-L data line. Logical '0' is the opposite. CAN-H is the dominant data line and CAN-L is the recessive data line. Often also a CAN-0 is present in the CAN-bus; it gives both data lines a common reference and thus makes the CAN-bus more stable.

The CAN-messages consist of an identifier ('address') and data. The messages are up to approximately 100 bits long and are send at a speed of 125.000 bits per second across the CAN-bus. So, every bit will be approximately 8 microseconds long and which means that any disturbance on the cable may cause a logical '1' to turn in a logical '0' (or the opposite) and thus cause an error in the message. Fortunately, the CAN bus has a very good error detection, but it may be clear that good CAN-bus connections are important for proper operation.

Due to the high-speed transmissions, the performance of the connection in the frequency domain is important. The impedance of the CAN-cable must be approximately 120 Ohms. Impedance is a kind of resistance of the cable for frequencies. When the resistance of a cable is not constant, this will cause reflections on the bus in the reverse direction. Hence, the CAN-bus must have a single line topology with terminating resistors (R) of 120 Ohms on both ends. These resistors simulate an endless continuation of the CAN-bus. A 'stub' as shown in the figure acts as an extra load and therefore should be avoided or kept as short as possible (i.e., less than 0.3m).

CAN-bus

Since messages are sent from one node to another node it is important that each node has a unique address. If not, then two nodes may reply simultaneously to a message causing a collision of transmissions and possible errors.

CAN-communication is done by Service Data Objects (SDO) and Process Data Objects (PDO). The SDOs are used for initialization and the adjustments of parameters. The PDO's are used for a more efficient data exchange, focussing on the process variables (e.g. inputs and outputs). In the standard CANopen standard, four TPDO's and four RPDO's are declared. The data length of the PDO's is limited to eight bytes. The PDO mapping describes the PDO objects the device sends and listens to during normal operation. Receiving of PDO's is guarded. If the communication is absent for 2 seconds, all outputs are switched off. A description of the PDO's can be found in the CUBIx-CAN manual, which is also located on the Kwant Control website under *Frequently Asked Questions*.