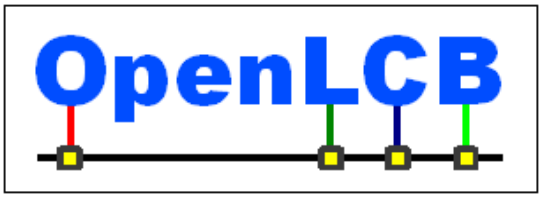

OpenLCB Technical Note	
Protocol Identification Protocol	
10/08/10	Preliminary

1 Introduction

OpenLCB defines a number of specific protocols for interacting with nodes, including event exchange, datagrams, streams, configuration, etc. These protocols are optional, in the sense that not every node will implement every one.

To determine which protocols a node implements, a Protocol Support Inquiry message is sent to the specific node. It will reply with a Protocol Support Reply message that contains six bytes of data. A specific bit position has been reserved for each defined protocol. If the bit is zero or not present, the protocol is not supported and requests to use it will result in a error. If present and 1, the protocol is supported.

It is not necessary to check whether an addressed protocol is supported by a node before attempting to use the protocol. If it's not, the standard error handling mechanism will indicate that.

This protocol provides a way to check, without errors, whether the protocol is supported. Avoiding errors provides a cleaner system. Further, this protocol can check support for protocols that use global (non-addressed messages); nodes are not permitted to return errors for global messages.

Generally, the node designer will just provide a simple fixed value for the reply.

2 Annotations to the Standard

2.1 Introduction

2.2 Intended Use

2.3 Reference and Context

2.4 Messages

The CAN messages were defined to be part of the “simple” subset. Low-end nodes may want to implement this protocol so that higher-function nodes can easily learn their limitations.

2.5 Interactions

Other nodes can snoop on these interactions to learn the protocols supported by a node, without having to send their own inquiry. It's not expected that the protocols supported by a specific node will change with time, although the Standard does not require that they be immutable.

2.6 Protocol Identification Values

The length of the field is constrained by the CAN packet length.

OpenLCB is big-endian, so these have been assigned from the MSB of the 1st byte.

The MSB for Protocol Identification Protocol is always 1, because it has to be present for the reply to be generated.

0x00 00 00 00 00 08 through 0x00 00 00 00 00 01 are reserved for control of future expansion. They'll eventually be needed when there are more than 44 protocols defined.

Table of Contents