OpenLCB / NMRAnet S-9.7
for Developers and DIY

OpenLCB

NMRAnet

GRAND RAILS 2012

Wednesday, 8 August 12
OpenLCB / NMRAnet
Technical Overview

• The OpenLCB Stack
• CAN Implementation
• Transport Protocols
• Application Protocols
Assumptions

• Have a vague sense of what OpenLCB is all about
• Understand Producer-Consumer model
Developers’ Tools

• Reference Hardware
• Embedded Software
• iPhone / PC Software
OpenLCB Stack

- Physical interface
- Data link layer
- Message / Network layer
- Event/Datagram/Stream Transport
- Application Layer
Physical Interface

- CAN interface defined by NMRA S-9.7.1
- RJ45 + CAT3 (or better)
Unique Identifiers

• Each node is permanently assigned a globally unique 6-byte NodeID
  • AA.BB.CC.DD.EE.FF
• Individuals, organizations, and manufacturers may apply for blocks
• Error for two nodes to share an ID
**Data Link Layer**

- Protocol is specified in abstract terms
- Can be implemented with nearly any communications technology
- CAN Implementation is complete
  - 29-bit header
  - 0–8-byte payload
Data Link Layer

• 29-bit header format:
  • 1-bit priority/frame type
  • 1-bit reserved
  • 3-bit message type
  • 12-bit MTI or destination alias
  • 12-bit source alias
**Alias?**

- Tricky to fit a 6-byte NodeID in a 29-bit header
- Trickier to fit two NodeIDs
- Over individual CAN segments, Nodes acquire a dynamic 12-bit alias
- Robust protocol to ensure uniqueness
- Transparent to higher levels
Data Link Layer

• Two Frame Types
  • CAN
    • Used mainly for alias allocation
  • OpenLCB
Data Link Layer

- CAN Message Types
- Check ID (CID)
- Reserve ID (RID)
- Alias Map Definition (AMD)
- Alias Map Enquiry (AME)
- Alias Map Reset (AMR)
Data Link Layer

• OpenLCB Message types:
  • Global Message
  • Addressed Message
  • Datagram Fragment (only / first / middle / last)
  • Stream Fragment
## Data Link Layer

- **MTI format:**

<table>
<thead>
<tr>
<th>bits</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:11</td>
<td>Priority</td>
</tr>
<tr>
<td>5:9</td>
<td>Type within Priority</td>
</tr>
<tr>
<td>4</td>
<td>Simple Protocol</td>
</tr>
<tr>
<td>3</td>
<td>Dest Alias Present in Payload</td>
</tr>
<tr>
<td>2</td>
<td>EventID present in Payload</td>
</tr>
<tr>
<td>0:1</td>
<td>Type Modifier</td>
</tr>
</tbody>
</table>
# Data Link Layer

<table>
<thead>
<tr>
<th></th>
<th>11000xxxxxxxxxxxxxxx[src 11:0]</th>
<th>0-8 bytes data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global Message</td>
<td>11001[MTI 23:12][src 11:0]</td>
<td>0-8 bytes data</td>
</tr>
<tr>
<td>Addressed Message</td>
<td>11001[MTI 23:12][src 11:0]</td>
<td>Dest Alias + 0-6 bytes data</td>
</tr>
<tr>
<td>Datagram-only</td>
<td>11010[dest 23:12][src 11:0]</td>
<td>0-8 bytes data</td>
</tr>
<tr>
<td>Datagram-first</td>
<td>11011[dest 23:12][src 11:0]</td>
<td>0-8 bytes data</td>
</tr>
<tr>
<td>Datagram-middle</td>
<td>11100[dest 23:12][src 11:0]</td>
<td>0-8 bytes data</td>
</tr>
<tr>
<td>Datagram-final</td>
<td>11101[dest 23:12][src 11:0]</td>
<td>0-8 bytes data</td>
</tr>
<tr>
<td>Reserved</td>
<td>11110xxxxxxxxxxxxxxx[src 11:0]</td>
<td>0-8 bytes data</td>
</tr>
<tr>
<td>Stream</td>
<td>11111[dest 23:12][src 11:0]</td>
<td>0-8 bytes data</td>
</tr>
</tbody>
</table>
Message / Network Layer

- Initialization Complete
- Verify NodeID Number (Global/Addressed) (VerifyNID)
- Verified NodeID Number (VerifiedNID)
- Optional Interaction Rejected (OIR)
- Terminate Due to Error (TDE)
Event Transport

- Event Reports announce state changes
- Produced in response to a layout event
- Trigger a consumer to take some action
- Contain a 8-byte EventID
- Most fundamental interaction
Event Transport

• Producer/Consumer Event Report (PCER)
• Learn Event
• Identify Consumer/Producer
• Identify Consumer/Producer Range
• Consumer/Producer Identified
• Identify Events (Global/Addressed)
Datagram Transport

- Datagrams are addressed
- Datagrams contain up to 72 bytes of payload
- Datagrams longer than 8 bytes must be split across multiple CAN frames
Datagram Transport

- Datagram (only)
- Datagram (first fragment)
- Datagram (middle fragment)
- Datagram (final fragment)
- Datagram Reply
Datagram Transport

- Received OK
- Rejected
  - Permanent Error
  - Datagram Type Not Accepted
  - Buffer Full (resend datagram)
  - Out of Order (resend datagram)
Stream Transport

- For moving large quantities of data
- Recipient has flow control
- Not meant for real-time streaming (e.g. live audio)
Stream Transport

- Stream Initiate Request
- Stream Initiate Reply
- Stream Data Send
- Stream Data Proceed
- Stream Data Complete
APPLICATION LAYER

• Protocol Inquiry Protocol (PIP)
• Memory Configuration Protocol
• Configuration Description Information (CDI)
• Simple Node Identification Protocol (SNIP)
• Ident Method
• Remote Button Protocol
• Display Protocol
• Traction Control Protocol
• &c.
Memory Configuration Protocol

• Primary method for node configuration
• Uses datagrams to read and write to node memory
• 255 distinct address spaces available
• 32-bit addresses
• 1,099,511,627,520 distinct addresses
• 1 terabyte of addressable memory
Configuration Description Information

- Describes the memory layout
- Provides human readable descriptions of configuration options
- Tells configurators how to read/write those options
- Contained in static XML that can be read directly from the node
Simple Node Identification Protocol

- Fast way to browse contents of NMRAnet
- Provides:
  - Read-Only manufacturer name and description
  - Read/Write user name and description
Configuration Tools

- JMRI currently supports:
  - SNIP
  - CDI
  - Memory Configuration
What We’re Working On

- Zigbee implementation
- Better configuration tools
- NMRAnet-based bootloaders
- Traction Control Protocol
- And recommended practices for interfacing with DCC
- Topology Autodiscovery
- For piecing together modular layouts on the fly
Useful Links
QUESTIONS?