MultiNode DLT

April 19, 2018 | Collect Data from Multiple Car Software Domains

Christoph Lipka
Senior Engineer, Advanced Driver Information Technology

This work is licensed under a Creative Commons Attribution-Share Alike 4.0 (CC BY-SA 4.0)
GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.
Copyright © GENIVI Alliance 2018.
Agenda

• Overview DLT
• MultiNode DLT
• What‘s next?
Overview DLT
DLT Overview

• Diagnostic Log and Trace
  – component for logging and tracing in ECUs in AUTOSAR 4.0
  – Provides standardized interface for logging and data format for transferring to host
  – Goals
    • Validation during development
    • Analysis of the end product
    • Standardization
  – Information and code
    • https://github.com/GENIVI/dlt-daemon

• DLT Viewer
  – ready to use host tool to view logs and to control the DLT Daemon and applications
  – Information and code
    • https://github.com/GENIVI/dlt-viewer
DLT Overview

- Supported Features
  - Support for multiple applications with multiple contexts organized in a hierarchy
  - Support for different interfaces between daemon and viewer (TCP/IP, Serial)
  - Verbose and Non-verbose mode logging
  - Predefined control messages
  - Message injection callback
  - User library and daemon provides a temporary internal buffer
  - Adapters to connect Linux log facilities like systemd-journal
  - MultiNode support
  - Offline Logging
  - etc.
DLT Overview

• Supported Features
  – Support for multiple applications with multiple contexts organized in a hierarchy
  – Support for different interfaces between daemon and viewer (TCP/IP, Serial)
  – Verbose and Non-verbose mode logging
  – Predefined control messages
  – Message injection callback
  – User library and daemon provides a temporary internal buffer
  – Adapters to connect Linux log facilities like systemd-journal
  – MultiNode support
  – Offline Logging
  – etc.
DLT Overview

- DLT Example Application

```c
#include <dlt/dlt.h>

DLT_DECLARE_CONTEXT(ctx); /* declare context */

int main()
{
    DLT_REGISTER_APP("MAPP", "Test Application for Logging"); /* register application */
    DLT_REGISTER_CONTEXT(ctx, "TES1", "Test Context 1 for Logging"); /* register context */
    DLT_LOG(ctx, DLT_LOG_ERROR, DLT_INT(5), DLT_STRING("This is an error")); /* Write your logs */

    /* ... */

    DLT_UNREGISTER_CONTEXT(ctx); /* unregister your contexts */
    DLT_UNREGISTER_APP(); /* unregister your application */
    return 0;
}
```
DLT Overview

- DLT Example Application in DLT Viewer
MultiNode DLT
New Use Cases

- **Container - 2 different scenarios**
  1. Applications inside container shall directly log messages to Gateway DLT Daemon
  2. Another DLT Daemon runs inside the container without direct connection to the outside world

- **Solution (1): Unix Sockets**
  - Use Unix sockets instead of named pipes for communication
    - [https://github.com/GENIVI/dlt-daemon/pull/43](https://github.com/GENIVI/dlt-daemon/pull/43)

- **Solution (2): MultiNode configuration**
New Use Cases

- ECU consolidation
  - DLT is deployed into different OSes but only one single connection to the outside
- Solution: MultiNode configuration
DLT MultiNode - Summary

• Allows to connect DLT Daemons running in different execution environments

• Any DLT Client can only be connected to the Gateway DLT Daemon

• All other passive DLT Daemons don’t need any physical connection to any DLT client

• All messages are routed through the Gateway DLT Daemon
  – From passive DLT Daemons to the DLT Client
  – Control messages from DLT Client to passive DLT Daemons

• Gateway DLT Daemon behaves like a DLT Client for passive DLT Daemons – therefore no change needed
DLT MultiNode - Configuration

• Precondition
  – Each DLT Daemon needs its own ECU Identifier
  – TCP connection between ECUs

# Set ECU ID (Default: ECU1)
ECUId = ECU1
DLT MultiNode - Configuration

- Configuration
  - Enable Gateway mode in dlt.conf for the DLT Daemon that acts as gateway

  # Enable Gateway mode (Default: 0)
  GatewayMode = 1

- Gateway configuration file

  [PassiveNode1]

  IPaddress=192.168.2.11
  ; TCP port. Default 3490 is used if no port is specified.
  Port=3490
  ; passive nodes ECU identifier. Mandatory.
  EcuID=ECU2
  ; Try connect to passive Node on DLT Daemon startup. Default OnStartup if not specified.
  Connect=OnStartup
  ; Stop connecting to passive node, if not successful after 10 seconds
  Timeout=10
  ; Send following control messages after connection is established
  SendControl=0x03,0x13
DLT MultiNode and Logstorage

- **Use case**: Collecting logs during test drives
- **Supported features**
  - Store application log messages to different storage devices available in the platform (e.g. USB, SD-Card, FLASH)
  - Runtime configuration
    - Start/Stop logging
    - Many configuration options per filter configuration

- **Related apps**
  - Audio app
  - Related apps
  - Related apps

- **Logstorage**
  - Configuration file
  - XXX.dlt
  - YYY.dlt

- **DLT-Viewer**
  - Choppy sound

15 | April 19, 2018 | Copyright © GENIVI Alliance 2018
**DLT MultiNode and Logstorage**

- **Use case**: Collecting logs during test drives
- **Supported features**
  - Store application log messages to different storage devices available in the platform (e.g. USB, SD-Card, FLASH)
  - Runtime configuration
    - Start/Stop logging
    - Many configuration options per filter configuration
What’s next?
What’s next?

- DLT POSIX compliance… first patches coming soon …
  - To run same DLT on other (RT)OS and easily deploy DLT MultiNode, some adaptions are needed
  - Refactor parts of the code to remove Linux specific system calls
  - High demand from users: [https://github.com/GENIVI/dlt-daemon/issues/52](https://github.com/GENIVI/dlt-daemon/issues/52)
What’s next?

- DLT and Security
  - We need a way to configure a filter for logs and control messages based on state on the system
    - During development -> everything allowed
    - At the customer -> only certain logs of applications, no TCP connection
    - During service -> a certain set of applications will log to internal logstorage, only certain control messages
  - Ideas/Requirements/Contributions are welcome
What’s next?

• Log consolidation
  – Not all execution environments use DLT logging (e.g. Android)
  – How to combine logs to get a (more) complete picture?
  – Ideas/Requirements/Contributions are welcome -> discussion point for SHDA project
Thank you!

Visit GENIVI at http://www.genivi.org or http://projects.genivi.org
Contact us: help@genivi.org