FlexRay Protocol and Architectures

Chris Quigley

Warwick Control Technologies
Aim: Give an overview of FlexRay protocol and network architectures

1. FlexRay properties
2. FlexRay Frame
3. FlexRay Timing
4. FlexRay Consortia
BMW FlexRay Business Case

• Recognised that NEW technology with HIGHER performance will result in higher nodal costs

• Nodal costs can be reduced in future by:-
  1. Using microcontroller with integrated FlexRay controller
  2. Using a low cost transceiver

• However on a system wide basis, cost savings include:-
  1. Replacing several CAN buses and their cables
  2. Reducing number of gateways
  3. Reduction of design effort for system partitioning
Network Architecture of Future

- Many proposed uses of FlexRay

- **FlexRay**
- High speed backbone
- X-by-Wire
- Airbag deployment
- LIN Sub Bus:
  - Doors
  - Seats etc.
- CAN/TTCAN – Applications:
  - Powertrain/body
  - TTCAN deterministic powertrain
  - MOST Infotainment
FlexRay Main Features (now 2.1)

- Multi-Master
- Bit rate:
  - Maximum Bit rate of 10Mbit/s
- Network Topologies:
  - Star, Bus, Mixed
- Physical Layer:
  - Twisted pair - electrical
- Safety:
  - Dual Channel redundancy
  - Bus Guardian for Babbling Idiot avoidance
FlexRay Frame Format

FlexRay frame format.
FlexRay Frame

• Header Segment:-
  • Control Bits
    • inc. Sync Bit
  • Frame ID (11 bits) – Analogous to CAN ID
  • Length (7 bits) – Length of payload in word (0 to 127 words = 0 to 254 bytes)
  • Header CRC (11 bits) – Protects header
  • Cycle (6 bits)

• Payload Segment:-
  • Message Data (0 to 254 bytes)
    • 1st two bytes can be used for Message ID
    • Can be used to Transpose J1939 PGN values

• Trailer Segment:-
  • CRC (24 bit)
TDMA in FlexRay

- Cycle subdivide into a static and a dynamic segment. Exclusive bus access enabled for short time in each case. Dynamic segment for transmission of variable length information. Bandwidth used when it is actually needed.
Structure of Flexray networks

- Bus guardian protects the system against failing processors, e.g. so-called “babbling idiots”
Node Architecture - Bus Guardian

- **BD** – Bus Driver
  - Electrical Physical layer
- **BG** – Bus Guardian
  - Protects message schedule
  - Stops “Babbling Idiot” failure

Bus Guardian application
FlexRay Network Topologies

Numerous topologies, include:-

• Passive Star:-
  • Low cost star
• Active Star:-
  • Fault tolerant star
• Linear Passive Bus:-
  • Similar to current CAN bus
• Dual Channel Bus:-
  • Dual redundancy
• Cascaded Active Star:-
  • Multiple couplers

• Dual Channel Cascaded Active Star:-
  • Additional safety
• Mixed Topology Network:-
  • Mixture of Star and Bus topologies
Dual Channel Bus

Dual channel bus configuration.
Message routing results in lower loading on some branches.
Cascaded Active Star Network

Example of a network with cascaded active stars
Mixed Channel Network

Example of a mixed channel network
Mixed Topology Network

Example of a mixed topology network
<table>
<thead>
<tr>
<th>Network</th>
<th>Bus</th>
<th>Multiple Star</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single Channel</strong></td>
<td>passive medium, most experience, cost efficient</td>
<td>allows for high data rates, increases error containment</td>
</tr>
<tr>
<td><strong>Dual Channel</strong></td>
<td>Electrical &amp; optical physical layer</td>
<td></td>
</tr>
</tbody>
</table>
Network topology – passive bus

- Maximum 8 stubs in a 10MBit/s network
- Maximum 0.2m length of stub
- Maximum 12m cable length between nodes
- Acceptable EMC performance achievable
- All in all feasible, but NO fault containment
Network topology – active star

- Maximum 16 branches
- Maximum 24m length of branch (without stubs)
- Maximum 12m length of branch (with stubs)
- Improved EMC performance
- Fault containment on branches

Short on bus wires
Network topology – active star

- Active stars can be cascaded, which means connected to each other with a point-to-point connection.
- No stub nodes on the connection between two active stars.
- A communication element that is sent from a node M to a node N may pass at most 2 active stars on its way.
Timing Hierarchy

Protocol timing related to the schedule of the communication cycle

communication cycle level

static segment

dynamic segment

symbol window

network idle time

arbitration grid level

static slot

minislot

macrotick level

action point

macrotick

microtick level

microtick
Timing Hierarchy

Four levels on an internal node basis

Communication Cycle Level - numbered from 0 to cCycleCountMax
Apart from startup, Communication Cycle is:-
• Executed periodically
• Consists of a constant number of Macroticks
• Contains the following components:-
  • Static Segment (Compulsory):- TDMA
  • Dynamic Segment (Optional):- Mini-slotting
  • Symbol Window (Optional) – Wake Up Symbol etc.
  • Network Idle Time (Optional)

Arbitration Grid Level
• Static Slot – Static Segment
• Mini-Slot – Dynamic Segment

Macrotick Level
Designated boundaries – action points – instants at which transmissions start

Microtick Level
Derived directly from the oscillator – a local node time – not used in synchronisation
Timing Hierarchy – Bit Times

Derived from clock ticks of the node’s oscillator
Message Exchange Options

Messages exchanged within frames

Frames sent within the recurring communication cycle

Frame scheduling performed autonomously by the protocol engine

Statically scheduled frames
• Repetitive message transfer with bounded communication latency
• Example: distributed control loops

Dynamically scheduled frames
• Spontaneous message transfer
• Example: Diagnostic Information
How is Synchronisation Achieved?

Some definitions

- Global time: Common time within a cluster or network
- Local time: the time of the nodes clock

Synchronisation is the minimisation of the difference between Global and Local time

- Achieved using Sync Frames in the Static Segment:
  - Sync Frame if Sync Bit is set
  - If set, receiving nodes should use for synchronisation

Sync frame configuration rule

- At least 3 nodes shall be configured to be Sync nodes -> Fault tolerance
- Sync nodes with 2 channels, sends one per channel
- Non-Sync nodes do NOT send Sync Frames
How is Synchronisation Achieved?

Time Measurement

- Every receiving node shall store (by channel), the time differences (in microticks) between the Local Time and the observed arrival time of all Sync frames.

External Clock Synchronisation

- Synchronisation of different FlexRay networks.
Bus Arbitration

Arbitration based on unique Frame Identifiers in Static and Dynamic segment

The Frame Identifier determines the transmission slot

Frame Identifier: 1 to SlotIdMax

Collision free arbitration via unique IDs and minislot counting
• Frame sent when scheduled frame ID matches slot counter
Statically Scheduled Frames

Frames of static length assigned uniquely to slots of static duration
- Frame sent when assigned slot matches slot counter

BG protection of static slots
Dynamically Scheduled Frames

Dynamic bandwidth allocation
• per node as well as per channel

Collision free arbitration via unique IDs and minislot counting
• Frame sent when scheduled frame ID matches slot counter

No BG protection of dynamic slots
Dynamically Scheduled Frames

Duration of Dynamic Slot depends upon whether or not frame tx or rx takes place

Each mini slot contains an Action Point (macroticks) when transmission takes place

If transmission does not take place, then moves to next mini-slot
**Communication Example**

**Communication Cycle Length**

<table>
<thead>
<tr>
<th>Cycle 0</th>
<th>Static Segment</th>
<th>Dynamic Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Static Slot 0</strong></td>
<td><strong>Dynamic Slot ID</strong> $m$</td>
</tr>
<tr>
<td></td>
<td><strong>Static Slot 1</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cycle 1</th>
<th>Static Segment</th>
<th>Dynamic Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Static Slot 0</strong></td>
<td>$m$</td>
</tr>
<tr>
<td></td>
<td><strong>Static Slot 1</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cycle 2</th>
<th>Static Segment</th>
<th>Dynamic Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Static Slot 0</strong></td>
<td>$m$</td>
</tr>
<tr>
<td></td>
<td><strong>Static Slot 1</strong></td>
<td></td>
</tr>
</tbody>
</table>

*Back to Cycle 0 again*
Network topology – cable termination

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>Resistor</td>
<td>&lt;10</td>
<td>Ohm</td>
</tr>
<tr>
<td>$C_1$</td>
<td>Capacitor</td>
<td>47</td>
<td>nF</td>
</tr>
</tbody>
</table>

$|R_T-Z_0|/Z_0 < 10\%$

Matching of termination

Bus Driver (at end node)

Split termination

STP or UTP cable

Network topology – cable termination

Networking Cables

Unshielded twisted-pair cable

Shielded twisted-pair cable

Coaxial cable

http://www.computerhope.com
Network topology – cable termination: better ESD protection

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{BP}$</td>
<td>Capacitance of BP to GND</td>
<td>$&lt; 50$</td>
<td>pF</td>
</tr>
<tr>
<td>$C_{BM}$</td>
<td>Capacitance of BM to GND</td>
<td>$&lt; 50$</td>
<td>pF</td>
</tr>
<tr>
<td>$C_{Diff}$</td>
<td>ECU's differential input capacitance</td>
<td>$&lt; 40$</td>
<td>pF</td>
</tr>
</tbody>
</table>

Hint: Capacitances to be measured at test frequency $f_{test} = 5MHz$
### Network topology – cable termination

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{\text{CMC}}$</td>
<td>Length resistance</td>
<td>$&lt; 300$</td>
<td>mOhm</td>
</tr>
<tr>
<td>$L_{\text{CMC}}$</td>
<td>Main inductance</td>
<td>Application specific selection necessary.</td>
<td>uH</td>
</tr>
<tr>
<td>$L_{\sigma}$</td>
<td>Stray inductance</td>
<td>$&lt;&lt; 1$</td>
<td>uH</td>
</tr>
</tbody>
</table>

#### Diagram

- **Bus Driver** (at end node)
  - **Split termination**
  - **STP or UTP cable**
  - **common mode choke**

- **BP**
- **BM**

- **To improve the emission & immunity performance**
- **Forces the current to be of the same strength, but opposite direction.**
- **The parasitic stray inductance should be as low as possible (keeps bus oscillations low).**
CAN Signals

Bit Convention
• Logical “0” = Dominant
• Logical “1” = Recessive

ISO 11898 CAN High Speed

<table>
<thead>
<tr>
<th>Recessive</th>
<th>Dominant</th>
<th>Recessive</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 V</td>
<td></td>
<td>0 V</td>
</tr>
<tr>
<td>2.5 V</td>
<td></td>
<td>V_{diff}</td>
</tr>
<tr>
<td>1.5 V</td>
<td></td>
<td>2 V</td>
</tr>
</tbody>
</table>

CAN_High

CAN_Low
CAN Physical Signaling

CAN Chip

CAN Driver

CAN Bus

Can_H

Can_L

CAN Signalling

1 = Recessive (Vdiff = 0V)
2 = Dominant (Vdiff = 2V)
**Electrical Signaling**

- Differential voltage \( u_{Bus} = u_{BP} - u_{BM} \)
- *Idle-LP* is when no current is drawn, i.e. Power Off situation. BP and BM are at GND.
- *Idle* is when no current is drawn but BP and BM are biased to the same voltage level.
- *Data_1*, BP is at a +ve level, BM is at a -ve level, Differential = +ve.
- *Data_0*, BM is at a +ve level, BP is at a –ve level, Differential = -ve.
# FlexRay Differential Voltages

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$uBDT_x$</td>
<td>Absolute differential voltage, while sending</td>
<td>600</td>
<td>2000</td>
<td>mV</td>
</tr>
<tr>
<td>$uBDT_x$</td>
<td>Absolute differential voltage, while Idle</td>
<td>0</td>
<td>30</td>
<td>mV</td>
</tr>
<tr>
<td>$iBP_{GNDShortMax}, iBM_{GNDShortMax}$</td>
<td>Maximum output current when shorted to GND</td>
<td></td>
<td>100</td>
<td>mA</td>
</tr>
<tr>
<td>$iBP_{BATShortMax}, iBM_{BATShortMax}$</td>
<td>Maximum output current when shorted to 14V</td>
<td></td>
<td>-100</td>
<td>mA</td>
</tr>
<tr>
<td>$dBDT_x10$</td>
<td>Transmitter delay, negative edge</td>
<td>100</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$dBDT_x01$</td>
<td>Transmitter delay, positive edge</td>
<td>100</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$</td>
<td>dBDT_x10 - dBDT_x01</td>
<td>$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$dBusTx10$</td>
<td>Fall time differential bus voltage (300mV → -300mV)</td>
<td>5</td>
<td>20</td>
<td>ns</td>
</tr>
<tr>
<td>$dBusTx01$</td>
<td>Rise time differential bus voltage (-300mV → 300mV)</td>
<td>5</td>
<td>20</td>
<td>ns</td>
</tr>
<tr>
<td>$dTx0$</td>
<td></td>
<td>0.95</td>
<td></td>
<td>gdBit</td>
</tr>
<tr>
<td>$dTx1$</td>
<td></td>
<td>0.95</td>
<td></td>
<td>gdBit</td>
</tr>
</tbody>
</table>
Voltage Levels

- The FlexRay PL has a buffer supplied by VBuf (typically around 5 volts)
- The idle level is half VBuf
  - Typically around 2.5 volts
- Red shows BP
- Green shows BM
Voltage Levels at Startup

- Shows rise from Idle_LP to Idle
Signal Integrity
3. Related Consortiums, Standards and Industry Initiatives
FlexRay Consortium

- Started 2001
- Core Members
  - BMW
  - Bosch
  - DaimlerChrysler
  - Freescale (formally Motorola)
  - GM
  - Philips
  - VW
- WCT – Development Member
ASAM

Association for Standardisation of Automation and Measuring Systems

Members

- Automotive OEMs
- 1st Tier suppliers
- Tool suppliers

www.asam.net

Key standards associated with FlexRay

- XCP – Calibration Protocol, where “X” denotes any protocol
- FIBEX – Equiv. CANdb or LIN Description File (LDF)
ASAM

XCP – Calibration Protocol

- Used for data logging and parameter tuning during development
- CAN, FlexRay and USB specified

FIBEX

- XML based file format for describing automotive control system
  - e.g.
    - Networks
    - Nodes
    - Messages
    - Signals (e.g. Engine Speed)
AUTomotive Open System ARchitecture, AUTOSAR

- WCT subcontractor advising on LIN software architecture
- Motivation: Basic I/O functionality is not a selling point for automotive OEM
  - therefore look to standardise software components
  - reduce costs of production of vehicle electronics
- Advantages:
  - Ease of portability between different microcontroller architectures
  - Open up the ability to use different software suppliers
AUTOSAR - Features

Goals

• Scalability to different vehicle and platform variants
• Implementation & standardization of basic system functions as an OEM wide “Standard Core“ solution
• Transferability of functions throughout network
• Integration of functional modules from multiple suppliers
• Maintainability throughout the whole “Product Life Cycle”
• Increased use of “Commercial off the shelf hardware”
• Software updates and upgrades over vehicle lifetime

Software Architecture

• AUTOSAR Software
• Runtime Environment
• Basic Software
JASPAR (Japan Automotive Software Platform Architecture)

- On-going Japanese consortium
- Similar aims to AUTOSAR
- Parts of JASPAR involved in AUTOSAR
- www.jaspar.jp
SAPECS
(Secured Architecture & Protocols for Enhanced Car Safety)

- **Capture Requirements of** :-
  - information around vehicle
  - telematic information between vehicle & infrastructure

- **FlexRay Demo**
  - Develop and integrate FlexRay IP for demo
  - Demo of power train control
  - Analysis / Qualification tool for displaying data

- **Qualification standards for systems**
  - Review of current
  - Suggestion of new procedures and tools for qualification
SAPECS
(Secured Architecture & Protocols for Enhanced Car Safety)

• Project Partners
  • Atmel: Project Lead
  • Valeo: Automotive applications expertise
  • CS: Vehicle and Telematic Information
  • AMIS: Semiconductor manufacturer
  • WCT: In-vehicle networking tools and expertise

• End date: June 2007
EASIS
Electronic Architecture and System Engineering for Integrated Safety Systems

- European consortium looking to propose future vehicle network topologies
- Partners include:
  - PSA
  - DaimlerChrysler
  - Volvo
  - TRW
  - etc.
EASIS – Functional Architecture

- **Global vehicle level**: Global control and supervision, e.g., integrated safety systems.
- **Inter-domain level**: Inter-domain control functions, e.g., integrated vehicle dynamics control.
- **Domain level**: Domain specific control functions, e.g., for powertrain or chassis.
- **Basic components**: Sensors and actuators, basic vehicle functions.

FlexRay™
The communication system for advanced automotive control applications.
Recommended Network Protocols

- **LIN** – used for the smart sensors/actuators connection in vehicle (sub-network)
- **CAN** (low speed) – used in the body domain
- **CAN** (high speed) – used in the powertrain domain
- **MOST** – used in the telematic domain
- **FlexRay** – currently not in series, designed for close-loop control and fault tolerant requirements, will typically be used in the chassis domain, or powertrain. Recommended for the Backbone in ISS.
EASIS – System Topology
EASIS: Low End Recommended Architecture

Diagram showing the architecture of EASIS with various components connected through communication protocols such as Diagnosis CAN (500KBit/s), Body CAN (125KBit/s), and FlexRay (10MBit/s). The diagram includes connections to Engine, Transmission, Brake, SWM, WSM, Radar (ACC), and Camera.
FlexRay Future – FlexRay II

FlexRay 2.1
• Dual channel
• Multi-master
• Large RAM requirements and no. gates – therefore expensive
• Implemented in BMW X5 chassis control (single channel) as high speed network NOT x-by-wire
• Audi to implement FlexRay in future

FlexRay II
• Aim to make FlexRay cheaper and ultimately replace CAN?
• Single-master
• Single channel
• Lower silicon requirements