Taking Full Advantage of the FlexRay Network for Motor Vehicles

Fujitsu Microelectronics America, Inc.

May 2006
Contents

Introduction .................................................. 2
FlexRay Advantages .......................................... 3
FlexRay Applications ........................................... 6
FlexRay Node Operations ..................................... 6
FlexRay Frames and Signals ................................. 8
FlexRay Solutions from Fujitsu Microelectronics America .......... 10

Introduction

The FlexRay networking standard for motor vehicles provides a foundation that will shape the control structure of automotive electronics for many years to come. FlexRay serves as the next step beyond CAN and LIN, enabling the reliable management of many more safety and comfort features. FlexRay suits X-by-Wire applications, for example.

This technology backgrounder offers an overview of FlexRay’s applications in motor vehicle networking and describes the network’s protocol, including the frame format, topology, bus signals, and node status transitions. Also included is a comparison of FlexRay and CAN.

This backgrounder concludes with a profile of FlexRay chips and development support from Fujitsu Microelectronics America. Based on a license from Bosch, Fujitsu has introduced a FlexRay starter kit and FlexRay controller ASSP.

FlexRay is a registered trademark of Daimler Chrysler AG. The FlexRay Consortium promotes the standardization of FlexRay as the next-generation in-car communication protocol. Fujitsu is an associate member of the FlexRay Consortium and an official member of AUTOSAR (an open-architecture partnership) and JasPar (Japan Automotive Software Platform and Architecture).
FlexRay Advantages

FlexRay focuses on a set of core needs for today’s automotive industry, including higher data rates than previous standards, flexible data communications, versatile topology options, and fault-tolerant operation.

FlexRay thus delivers the speed and reliability required for next-generation in-car control systems. The CAN network has reached its performance limits with a maximum speed of 1 Mbps. With a maximum data rate of 10 Mbps available on two channels, giving a gross data rate of up to 20Mbit/sec, FlexRay potentially offers 20 times higher net bandwidth than CAN when used in the same application.

FlexRay also offers many reliability features not available in CAN. Specifically, a redundant communication capability enables fully duplicated network configurations and schedule monitoring by hardware. FlexRay also offers flexible configurations, with support for topologies such as bus, star, and hybrid types (Figure 1). Designers can configure distributed systems by combining two or more of these topologies.

Moreover, FlexRay allows both synchronous (real-time) and asynchronous data transfer to meet the demand for various systems in vehicles. For example, a distributed control system usually requires synchronous data transmission.

To meet diverse communication requirements, FlexRay also provides both static and dynamic communication segments within each communication cycle. The static communication segment provides bounded latency, and the dynamic segment helps meet varying bandwidth requirements that can emerge at system run time. The fixed-length static segment of a FlexRay frame transfers messages with a fixed-time-trigger method, and the dynamic segment transfers messages with a flexible, time-trigger method.

In addition to operating as a single-channel system like CAN and LIN, FlexRay can operate as a dual-channel system. The dual-channel option makes data available via a redundant network—a vital capability for a high-reliability system.
As shown in Table 1, FlexRay’s characteristics suit real-time control functions. FlexRay offers the highest reliability among the protocols shown in the table. Figure 2 further compares networking standards by node cost and data rate. Table 2 gives a detailed comparison of FlexRay and CAN.

Table 1—Vehicle Network Standards

<table>
<thead>
<tr>
<th>Class</th>
<th>Communication Speed</th>
<th>Applications</th>
<th>LIN</th>
<th>CAN</th>
<th>FlexRay</th>
<th>MOST/1394</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>10K to 125Kbps (body)</td>
<td>Lamps, lights, power windows, door locks, power seats, etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class B</td>
<td>125K to 1Mbps (status information)</td>
<td>Electronic indicators, driving information, automatic air conditioner, failure diagnosis, etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class C</td>
<td>1M to 10Mbps (real-time control)</td>
<td>Engine control, ABS, transmission control, break control, suspension control, etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class D</td>
<td>10Mbps or faster (multimedia)</td>
<td>Car navigation system, audio system, etc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The arrows indicate the required data rate.
Table 2—FlexRay and CAN Comparison

<table>
<thead>
<tr>
<th>#</th>
<th>ITEM</th>
<th>CAN</th>
<th>FlexRay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baud rate</td>
<td>1 Mbps</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>2</td>
<td>Number of channel for one node</td>
<td>1 ch</td>
<td>2 / 1 ch (optional)</td>
</tr>
<tr>
<td>3</td>
<td>Network topology</td>
<td>Bus type</td>
<td>Mix. of bus and star type</td>
</tr>
<tr>
<td>4</td>
<td>Connection node (max.)</td>
<td>16 nodes at 500 Kbps</td>
<td>22 nodes (bus)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>22 / 64 nodes (star)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>64 nodes (mixed)</td>
</tr>
<tr>
<td>5</td>
<td>Physical layer</td>
<td>Metal</td>
<td>Metal / POF</td>
</tr>
<tr>
<td>6</td>
<td>Communication</td>
<td>Event triggered</td>
<td>Time triggered + event triggered</td>
</tr>
<tr>
<td>7</td>
<td>ID</td>
<td>11 / 29 bits</td>
<td>11 bits</td>
</tr>
<tr>
<td>8</td>
<td>Data length code (DLC)</td>
<td>8 bytes</td>
<td>254 bytes</td>
</tr>
<tr>
<td>9</td>
<td>Frame</td>
<td>Data frame, remote frame, error</td>
<td>Data frame</td>
</tr>
<tr>
<td></td>
<td></td>
<td>frame, overload frame</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Bus line lock</td>
<td>Dominant lock probable</td>
<td>Babbling idiot (support with BG)</td>
</tr>
<tr>
<td>11</td>
<td>Error status transition</td>
<td>Error active, error passive, bus off (software restoration possible)</td>
<td>Normal active, normal passive, halt</td>
</tr>
<tr>
<td>12</td>
<td>Error counter</td>
<td>Status transition counter value</td>
<td>Any status transition counter value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fixed</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Type of errors</td>
<td>Bit error, stuffing error, CRC</td>
<td>Clock sync. error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>error, framing error, ACK error</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Oscillator</td>
<td>Ceramic and/or crystal</td>
<td>Crystal oscillator (BG separated from CC clock)</td>
</tr>
<tr>
<td>15</td>
<td>Network management</td>
<td>Software</td>
<td>Hardware (controlled by BD and BG)</td>
</tr>
<tr>
<td>16</td>
<td>Network synchronization</td>
<td>Synchronization only with sync. seg</td>
<td>Rate compensation and offset compensation possible</td>
</tr>
<tr>
<td>17</td>
<td>Bus length</td>
<td>40 meters at 1 Mbps</td>
<td>22 meters (in an active star, and between active star)</td>
</tr>
</tbody>
</table>

Notes:
- Babbling Idiot: Incorrect transfer causing damage
- BG: Bus guardian
- CC: Communication controller
- BD: Bus driver
FlexRay Applications

FlexRay targets many X-by-Wire uses in automobiles, as shown in Figure 3. Also shown in the figure is a gateway that interfaces between FlexRay and CAN networks.

Examples of FlexRay X-by-Wire applications include:

- Steering-by-Wire—Typically using electronic control unit
- Anti-lock brake system (ABS)—Including vehicle stability control (VSC) and vehicle stability assist (VSA)
- Power train—Controlling an electronic throttle that replaces the current mechanical system. The electronic throttle works in conjunction with existing systems such as a computerized fuel injector, computerized variable intake control system, and computerized idling control system.

FlexRay Node Operation

Each FlexRay node consists of a controller part and a driver part (Figure 4). The controller part includes a host processor and a communication controller. The driver part typically includes bus drivers and bus guardians (optional). The bus driver connects the communication controller to the bus, and the bus guardian monitors access to the bus. The host informs the bus guardian which time slots the communication controller has allocated. The bus guardian then allows the communication controller to transmit data only in these time slots and enables the bus driver. If the bus guardian detects a gap in the timing, it disconnects the communication channel.
As shown in Figure 5, a FlexRay node has several basic operational states:

- **Configuration (default config/config)**—For making all kinds of initial settings, including the communication cycle and data rate
- **Ready**—For making internal communication settings
- **Wakeup**—For waking up a node that is not communicating. In this state, the node sends the wakeup signal to another node, which wakes up and enables the communication controller, bus driver and bus guardian.
- **Startup**—For starting clock synchronization and getting ready for communication
- **Normal (Active/Passive)**—Communication-available state
- **Halt**—For indicating that communication has stopped
FlexRay nodes also have state transitions related to error handling (Figure 6). These transitions are managed based on the value of error counters for clock synchronization and clock-correction errors. A clock-correction error occurs when an individual node clock differs from the FlexRay sync node clock. A FlexRay network has one or more sync nodes, which transmit sync messages. On reception of each sync message, a node compares its clock with that of the sync node clock and makes any changes needed to synchronize.

Each node keeps an error count that includes the number of successive failures in clock synchronization. A node also monitors errors regarding frame transmission/reception status, including syntax errors, content errors, bus-violation errors, and errors caused by transmission conflicts. When a node detects any of these errors, it notifies the host processor.

The use of the error counter values depends on the application and is determined during system design. Depending on the error condition, for example, a node can halt communication.

**FlexRay Frames and Signals**

FlexRay uses a communication frame that has three segments (Figure 7).

FlexRay uses a time-trigger protocol, in contrast to CAN's event-trigger protocol. FlexRay's time-trigger method enables accurate data transfers according to a predefined schedule. Additionally, the data is available on dual-redundant communication channels, Ach and Bch.
The header segment includes the following bits:

- **Reserved bit**—For future expansion.
- **Payload preamble indicator**—Indicates the existence of vector information in the frame’s payload segment. In a static frame, this bit indicates NWVector; in a dynamic frame, the bit indicates Message ID.
- **Null frame indicator**—Indicates whether the data frame in the payload segment is NULL.
- **Sync frame indicator**—Indicates that this is a synchronous frame.
- **Startup frame indicator**—Indicates whether the node sending the frame is the startup node.
- **Frame ID**—Indicates the ID assigned to each node during system design (valid range: 1 to 2047).
- **Length**—Specifies the data length of the payload segment.
- **Header CRC**—Specifies the CRC calculation values of the Sync Frame Indicator, Startup Frame Indicator, Frame ID, and Length that are calculated by the host.
- **Cycle**—Indicates the cycle count of the node that transfers the frame during the frame transfer time.

The frame’s payload segment includes these parts:

- **Data**—Can be from 0 to 254 bytes.
- **Message ID**—Optional. This ID uses the first two bytes of the payload segment for definition and can be used as filterable data on the receiving side.
- **Network management vector (NWVector)**—Optional. This vector must be 0 to 12 bytes long and common to all nodes.

The frame’s trailer segment consists of CRC values specified by hardware. These CRC values change the seed value on the connected channel to prevent incorrect connections.

FlexRay transfers frames in time slots. Figure 8 shows the organization of these time slots with regard to the FlexRay cycles.

![Figure 8—FlexRay Time Slots](image).

At the physical layer, FlexRay communicates using the differential signals BP and BM, corresponding to the voltages uBP and uBM. Four signals (Figure 9) represent various states of the FlexRay bus:

- **Idle_LP**: low-power state
- **Idle**: no-communication state
- **Data_1**: logical HIGH
- **Data_0**: logical LOW

Note that conflicts between Data_1 and Data_0 are not allowed.
FlexRay Solutions from Fujitsu Microelectronics America

After several years of refinement, the FlexRay standard is ready for system developers to leverage in next-generation vehicles. Fujitsu is supporting these efforts with a development system and a microcontroller containing FlexRay IP. Figure 10 shows the roadmap for these developments.
Fujitsu’s FlexRay Evaluation Kit (MB2005-01/FlexRay-FPGA-Eva-Kit-369) includes the FPGA-based ERAY IP from Bosch. This ERAY IP implements the latest FlexRay protocol at the time of shipment of the kit. Developers can use this kit with Fujitsu microcontrollers such as the 32-bit Fujitsu FR (MB91F369 MCU) starter kit.

![Figure 11—Evaluation Kit Boards](image)

As illustrated in Figure 11, the starter kit includes all the functionality required for FlexRay application development. Specifically, the kit contains:

- **Host processor board**
  - 32-bit MCU MB91F369
  - Two CAN interfaces, UART, SIOs
  - 1 MB flash (on board)
  - 512 Kbytes Flash (on chip)
  - Onboard monitor debugger
  - 2 MB SRAM (on board)
  - 32 Kbytes RAM (on chip)

- **FlexRay main and daughter boards with sockets for physical-layer modules**

- **Software**
  - Driver library for FlexRay interface
  - Example program

- **Development tools**
  - Softune Workbench on Micro CD 3.6 or later version
  - DECOMSYS tool chain demo CD

Fujitsu’s new FlexRay starter kit (SK-91F467-FLEXRAY) enables designers to evaluate the Fujitsu standalone FlexRay controller (MB88121A) along with the Fujitsu 32-bit flash microcontroller MB91F467DA. The kit includes an evaluation copy of the DECOMSYS::COMMSTACK driver library that allows easy access to the FlexRay communication controller MB88121A. The kit includes:

- 32-bit Flash microcontroller MB91F467DA
- FlexRay ASSP MB88121A
- Two FlexRay channels (Ch-A, Ch-B)
- FlexRay physical layer RS-485 on board
- FlexRay physical layer plug-in for PL modules from TZM (FT1080)
- 32 Mbit SRAM on-board memory
• Three high-speed CAN interfaces
• Three UART (RS232- or LIN-mode configurable)
• External bus interface on 96-pin/48-pin connectors (DIN 41612) for user applications
• Application examples on the FlexRay CD
• Softune Workbench development environment on CD
• DECOMSYS tool chain demo CD

Fujitsu also offers a FlexRay ASSP that supports FlexRay protocol version 2.1 (ES2). This ASSP provides the following features:

• QFP64 package
• 0.5-pin pitch/10 x 10 mm (M03-package)
• Single-supply voltage
• Clock specification, 4/5/8/10 MHz crystal
• Configurable parallel host interface compatible with 8-, 16-, and 32-bit MCUs. Max. frequency 33 MHz (target)
• Serial host interface (will be supported in the future devices)

In 2006, Fujitsu will introduce two FR series 32-bit microcontrollers with an integrated FlexRay macro. Figure 13 shows the features of the first of these MCUs, and Figure 14 shows a subsequent MCU with more on-board peripherals.
Taking Full Advantage of the FlexRay Network for Motor Vehicles

Figure 13—Fujitsu’s First FlexRay MCU (Plan)

Figure 14—Fujitsu FlexRay MCU with Additional Functionality (Plan)
For more information

Fujitsu Microelectronics America, Inc. (FMA) leads the industry in innovation. FMA provides high-quality, reliable semiconductor products and services for the networking, communications, automotive, security and other markets. For product information, visit the company web site at [http://www.fujitsu.com/us/services/edevices/microelectronics/microcontrollers/flexray/](http://www.fujitsu.com/us/services/edevices/microelectronics/microcontrollers/flexray/) or please address e-mail to inquiry@fma.fujitsu.com