FlexRay: A Time-Triggered Protocol for Drive-by-Wire Applications

Syed Masud Mahmud, Ph.D.
Electrical and Computer Engg. Dept.
Wayne State University
Detroit MI 48202
smahmud@eng.wayne.edu
Need for New Types of In-Vehicle Networks

- Drive-by-wire,
- Telematics,
- Entertainment,
- Multimedia,
- Pre-Crash Warning,
- Hit Avoidance,
- Remote Diagnostic
- Remote Software Update, etc.
Need for New Types of In-Vehicle Networks (contd..)

• Drive-by-Wire and Active Collision Avoidance Systems need *fault tolerant* networks with *time-triggered* protocols, to guarantee *deterministic latencies.*
Need for New Types of In-Vehicle Networks (contd..)

- Multimedia Systems need networks with *high bandwidth* to transfer video files
- Body control electronics need *low-bandwidth* networks to keep the cost down.
Need for Time Triggered Protocols
Average Latency and Maximum Allowed Latency

Maximum Allowed Latency for Safety Critical Applications

Number of Nodes

Average Latency

Flexray: A Time-Triggered Protocol
Latency Boundaries

Average Latency

Number of Nodes

Maximum Allowed Latency for Safety Critical Applications

Latency Boundaries
Failure of all Deadlines

All deadlines fail beyond this point.

Maximum Allowed Latency for Safety Critical Applications

Maximum Allowed Latency for Other Applications

Latency Boundaries

Average Latency

Number of Nodes

N_{\text{max}}
Need for Time Triggered Protocols

• Safety critical messages must meet their deadlines even at maximum bus load.
An Accurate Timekeeping Device is Needed for Time-Triggered Protocols

• In 1707, four British warships were lost at sea, and about 2000 sailors died because at that time people didn’t know how to calculate longitude accurately.

• In 1714, British government declared £20,000 (about $12 to $13 millions in 2020’s money) award to find a solution to the longitude problem.

• The Longitude Act of 1714
• John Harrison: The Accurate Clockmaker
• Longitude Movie: Part-1
• Longitude Movie: Part-2
Core Partners of the FlexRay Consortium

FlexRay: Basic Characteristics

- Synchronous and asynchronous frame transfer.
- Guaranteed frame latency.
- Prioritization of frames during asynchronous transfer.
- Multi-master clock synchronization.
- Error detection and signaling.
- Error containment on the physical layer.
- Scalable fault tolerance.
A Differential Voltage Link

BP: Bus Plus  BM: Bus Minus

Signal Voltage = Volt_{BP} - Volt_{BM}
**Idle_LP**: The Bus is in Idle Low Power Mode. Both BP and BM pins are at 0v.

**Idle**: The Bus is in Idle Mode. Both BP and BM pins are biased to a certain voltage.
Terminated Cable End

Terminated cable end.

© 2006 Syed Masud Mahmud
Un-Terminated Cable End

Un-terminated cable end.
Point to Point Terminated Connection

![Diagram of Point to Point Terminated Connection]

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBus</td>
<td>Length of a point-to-point connection</td>
<td>24</td>
<td>m</td>
<td></td>
</tr>
</tbody>
</table>
Network Topology

- Single Channel Passive Bus
- Dual Channel Passive Bus
- Single Channel Passive Star
- Dual Channel Passive Star
- Single Channel Active Star
- Dual Channel Active Star
- Various Combinations of Bus and Star Topologies
A Single Channel Passive Bus

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBus</td>
<td>Maximum electrical distance between two nodes in the system</td>
<td>24</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>IStubDistance_{M,N}</td>
<td>Distance between two network splices</td>
<td>150</td>
<td></td>
<td>mm</td>
</tr>
<tr>
<td>nStubNodes</td>
<td>Number of stub nodes</td>
<td>22</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A Dual Channel Passive Bus
A Single Channel Passive Star

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$IPassiveStar_{N+M}$</td>
<td>Maximum electrical distance between two nodes in the system</td>
<td>24</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>$nStarNodes$</td>
<td>Maximum number of nodes in a passive star network.</td>
<td>22</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A Single Channel Active Star

- **Node 1** connects to **Active Star** via **\( I_{ActiveStar_1} \)**
- **Active Star** connects to **Node 2** via **\( I_{ActiveStar_2} \)**
- **Active Star** connects to **Node 3** via **\( I_{ActiveStar_3} \)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_{ActiveStar_N} )</td>
<td>Length of a branch from node N to the star</td>
<td>24</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>( n_{ActiveBranches} )</td>
<td>Number of branches at an active star</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Internal Block Diagram of an Active Star with Four Branches
Internal Logic of an Active Star with Two Branches

Active Star

Transmitter

Receiver

Termination

Receiver

Transmitter

Termination
A Single Channel Cascaded Active Stars

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n\text{StarPath}_{M,N}$</td>
<td>Number of stars on the signal path from any node $M$ to a node $N$ in network with active stars</td>
<td>0</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>$l\text{StarStar}$</td>
<td>Electrical distance between two active stars</td>
<td>24</td>
<td>m</td>
<td></td>
</tr>
</tbody>
</table>
A Single Channel Hybrid Topology
A Dual Channel Active Star Network
A Dual Channel Cascaded Active Star Network
A Dual Channel Hybrid Network

Node A  Node B  Node C  Node D  Node E

Star 1B

Channel A

Flexray: A Time-Triggered Protocol

© 2006 Syed Masud Mahmud
Node Architecture

- Communication Data
- Configuration Data & Status Information
- Synchronization Signals
- Control Signals
- Status Signals
- Control Data & Status Information
- Control Signal (optional)
- Bus

Host

Communication Controller

Bus Guardian

Bus Driver

Power Supply

Wayne State University - College of Engineering
Host – Communication Controller Interface

- The host provides *control* and *configuration* information to the CC, and *provides payload data* that is transmitted during the communication cycle.
- The CC provides *status information* to the host and *delivers payload data* received from communication frames.
Bus Guardian

• The bus guardian (BG) has *a-priori* knowledge of the transmission times of the node.

• If the bus guardian detects any *mismatch* between the schedules of the communication controller and the bus guardian, it *signals an error* condition to the host and inhibits any further transmission attempts.
Host – Bus Guardian Interface

- The interface between BG and the host is a serial peripheral interface (SPI).
- This interface allows the host to send *configuration data* to the BG and receive *status information* from the BG.
CC–BG Interface

- This interface serves to *synchronize* the BG to the schedule of the CC.
- The CC also receives *synchronization information* back from the BG via the BGE signal.
Block Diagram of a Dual-Channel Node.
FlexRay Frame Format
Reserved Bit (1 bit):

- It is reserved for future use.
- A transmitting node shall set this bit to ‘0’.
- A receiving node shall ignore the reserved bit.
Header Segment (40 bits)

Payload Preamble Indicator (1 bit):
1: The payload segment contains a network management vector (NMV) in the static segment or a message ID (MID) in the dynamic segment.
0: The payload segment has no NMV or MID.
Null Frame Indicator (1 bit):

0: The payload segment contains no valid data. All bytes in the payload section are set to zero.

1: The payload segment contains valid data.
Flexray: A Time-Triggered Protocol

Header Segment (40 bits)

Sync Frame Indicator (1 bit):
0: No receiving node shall consider the frame for any type of synchronization.
1: All receiving nodes shall utilize the frame for synchronization if it meets other acceptance criteria.
Header Segment (40 bits)

Startups Frame Indicator (1 bit):

0: The frame is not a startup frame.

1: The frame is a startup frame. This bit shall only be set in the sync frames of coldstart nodes. Therefore, a frame with the startup frame indicator set to 1 shall also have the sync frame indicator set to 1.
Frame ID (11 bits)

- The frame ID defines the slot in which the frame should be transmitted. (Performs Arbitration)
- Each frame that may be transmitted in a cluster has a frame ID assigned to it.
- The frame ID ranges from 1 to 2047. The frame ID 0 is an invalid frame ID.
Payload Length (7 bits)

- The payload length indicates the number of 16-bit words in the payload segment.
- Thus, the payload segment can contain 0 to 127 words or **0 to 254 bytes**.
The Header CRC is computed over the Sync Frame Indicator, Startup Frame Indicator, Frame ID and Payload Length.
### Cycle Count (6 bits)

- This field indicates the current cycle number.
- A FlexRay communication cycle is similar to the TTCAN basic cycle.

<table>
<thead>
<tr>
<th>Field</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header CRC Covered Area</td>
<td></td>
</tr>
<tr>
<td>Reserved bit</td>
<td></td>
</tr>
<tr>
<td>Payload preamble indicator</td>
<td></td>
</tr>
<tr>
<td>Null frame indicator</td>
<td></td>
</tr>
<tr>
<td>Sync frame indicator</td>
<td></td>
</tr>
<tr>
<td>Startup frame indicator</td>
<td></td>
</tr>
<tr>
<td>Frame ID</td>
<td>11</td>
</tr>
<tr>
<td>Payload length</td>
<td>7</td>
</tr>
<tr>
<td>Header CRC</td>
<td>11</td>
</tr>
<tr>
<td>Cycle count</td>
<td>6</td>
</tr>
</tbody>
</table>
For frames transmitted in the static segment the first 0 to 12 bytes of the payload segment may optionally be used as network management vector.
Payload in Dynamic Segment

- For frames transmitted in the dynamic segment the first two bytes of the payload segment may optionally be used as a message ID field, allowing receiving nodes to filter or steer data based on the contents of this field.
Trailer Segment: Frame CRC

- The Frame CRC field is computed over the header segment and the **payload segment** of the frame.
Trailer Segment: Frame CRC

- The frame CRC has a Hamming distance of **six** for payload lengths up to and including 248 bytes.
- For payload lengths greater than 248 bytes the CRC has a Hamming distance of **four**.
Communication Cycle

One Communication Cycle

Static Segment
- For Time-Triggered Messages

Dynamic Segment
- For Event-Triggered Messages

Symbol Window
- For Special Signals such as Wake Up

Network Idle Time
- It allows time to do clock corrections and housekeeping operations at a node
Communication Cycle

One Communication Cycle

Static Segment | Dynamic Segment | Symbol Window | Network Idle Time

Similar to the Exclusive Window of the TTCAN system

Similar to the Arbitrating Window of the TTCAN System
Timing Hierarchy

communication cycle level

macrotick level

microtick level

vCycleCounter

gdCycle

vMacroTick

gdMacroTick

cCycleCountMax

gMacroPerCycle - 1

vMicroTick

vMicroTick

pdMicroTick

Flexray: A Time-Triggered Protocol

© 2006 Syed Masud Mahmud
Microtick

- **Microticks** are local time units of a node.
- The granularity of a node’s local time is a microtick.
- Microticks may have different durations in different nodes. (*For example, in one node a microtick may be 10 ns and in another node it may be 5 ns.*)
- The microtick is derived directly from the node’s oscillator clock tick, optionally making use of a prescaler.
Macrotick
(Similar to the NTU of the TTCAN System)

- The duration of a *macrotick* is equal throughout all synchronized nodes in a cluster.
- Each local macrotick is an integer number of local microticks.
- The number of microticks per macrotick may vary within the same node due to rate corrections.
- The number of microticks per macrotick may also differ between nodes, and depends on the oscillator frequency and the prescaler.
Length of a Communication Cycle (Cycle)

- A *cycle* consists of an integer number of macroticks.
- The number of macroticks per cycle shall be identical in all nodes in a cluster, and remains the same from cycle to cycle.
- At any given time all nodes should have the same cycle number.
An **action point** (AP) is a time instant at which a node performs a specific action, e.g. transmits a frame.
• The static segment contains a number of static slots.
• For a given system, the **number of static slots** in the static segment is a global **constant**.
• All static slots are of **same length** in terms of macroticks.
Structure of the Static Segment

- Every node maintains two *slot counters*: one for Channel A and one for Channel B.
- The node *simultaneously increments* both counters at the end of each static slot.

```
channel A

1---------------------- frame ID 1
2---------------------- frame ID 2
3

channel B

1---------------------- frame ID 1
2----------------------
3----------------------

slot counter channel A

slot counter channel B

static slot 1

static slot 2

static slot 3
```
Arbitration in the Static Segment

- **Static Slot “N”** is reserved for **Frame ID “N”**.
- If a node detects that its slot counters contain the same value as the Frame ID of one of its queued messages, then it sends that message.
- Static Slot “N” of the current cycle will *remain idle* if the node that contains Frame ID “N” has no message to send now.
Fault-Tolerant Transmission in the Static Segment

• If a node is connected to both channels, then the node can *simultaneously send* a time-triggered message through both channels.

• However, if a node is connected to only one channel (say Channel A), then Channel B will *remain idle* when the node sends a time-triggered message through Channel A.
Timing within the Static Segment

channel active | channel idle | delimiter | channel idle

frame ID 1

channel active | channel idle | delimiter | channel idle

frame ID 2

macrotick

static slot action point

action point offsetgdActionPointOffset

static slot gdStaticSlot

counter 1

2

microtick

slot 1

slot 2
Appropriate configuration of the static slot length \((gdStaticSlot)\) must assure that the frame and the channel idle delimiter and any potential safety margin fit within the static slot under worst-case assumptions.
Structure of the Dynamic Segment

- The dynamic segment contains a number of minislots.
- For a given system, the number of minislots in the dynamic segment is a global constant.
All minislots are of **same length** in terms of macroticks.

In the dynamic segment the **duration** of a communication slot depends upon the frame length.
Arbitration in the Dynamic Segment

• Each node continues to maintain the two slot counters – one for each channel.
• The slot counters are *incremented independently* according to the arbitration scheme within the dynamic segment.
• The slot number of the first minislot in the dynamic segment is the last static slot number plus 1.
Timing within the Dynamic Segment

- Channel active
- Channel idle
- Delimiter

Frame ID: m

Minislot
Minislot action
Point
Dynamic slot transmission phase
Dynamic slot idle phase
Dynamic slot
Slot counter: m

Frame ID: m+3

Minislot action
Point
Dynamic slot transmission phase
Dynamic slot idle phase
Dynamic slot
Slot counter incremented at the end of each dynamic slot

© 2006 Syed Masud Mahmud
Timing within the symbol window.

Arbitration is not provided by the protocol for the symbol window. The arbitration has to be performed by means of a higher-level protocol.
Network Idle Time

- During the network idle time a node calculates and applies clock corrections.
- The network idle time also serves as a phase during which the node performs implementation specific housekeeping operations.
- The network idle time contains the remaining number of macroticks within the communication cycle not allocated to the static segment, dynamic segment, or symbol window.
Frame Encoding in the Static Segment

- The transmitting node generates a **Transmission Start Sequence (TSS)** that consists of a continuous LOW for a given period.

- The **Frame Start Sequence (FSS)** consists of one HIGH bit. The **Byte Start Sequence (BSS)** is used to provide timing information to the receiver. The BSS consists of one HIGH bit followed by one LOW bit.

- The BSS is followed by eight data bits.

- The **Frame End Sequence (FES)** marks the last byte of a frame. The FES consists of one LOW bit followed by one HIGH bit.
Frame Encoding in the Dynamic Segment

- The **Dynamic Trailing Sequence** (DTS) is used to indicate the exact point in time of the transmitter’s minislot action point.
- After sending the FES, the transmitter keeps the TxD line low until the next minislot action point.
- The transmitter then moves the line to the HIGH level. (DTS → Variable number of LOW bits + one HIGH bit).
Sync Nodes and Sync Frames

- At least **three** nodes shall be configured to be sync nodes.
- Sync nodes must be connected to both channels.
- The sync frames shall be sent in the same static slot in each cycle.
- Non sync nodes must not transmit frames with the sync frame indicator set.
Rate Correction

- The Rate Correction determines by how many microticks the node’s cycle length should be increased or decreased.
- A node determines the length of a communication cycle by taking the difference between the arrival times of a pair of Sync Frames, with the same Frame ID, on consecutive cycles.
Rate Correction

- If there are two pairs for a given Sync Frame ID (one pair for channel A and another pair for channel B) the average of the differences is used.
- The process is repeated for all Sync Frames, and a fault-tolerant midpoint algorithm is executed for the correction.

\[
\text{Average Length} = \frac{(L_A + L_B)}{2}
\]
Offset Correction

• The offset correction value indicates how many microticks the node should shift the start of its cycle.

• Negative values mean the NIT should be shortened (making the next cycle start earlier). Positive values mean the NIT should be lengthened (making the next cycle start later).

Example: NIT has been shortened
Offset Correction

• A node determines the offset of a communication cycle by taking the difference (deviation) between the actual arrival times of a given Sync Frame ID and its expected arrival time.

• If a given Sync Frame ID has two deviation values (one for channel A and one for channel B) the smaller value is selected.

• The process is repeated for all Sync Frames, and a fault-tolerant midpoint algorithm is executed for the correction.
Fault-Tolerant Midpoint Algorithm

The algorithm determines the value of a parameter, k, based on the number of values in the sorted list.

<table>
<thead>
<tr>
<th>Number of values</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 2</td>
<td>0</td>
</tr>
<tr>
<td>3 – 7</td>
<td>1</td>
</tr>
<tr>
<td>&gt; 7</td>
<td>2</td>
</tr>
</tbody>
</table>

FTM term deletion as a function of list size.
Fault-Tolerant Midpoint Algorithm

- The measured values are sorted and the k largest and the k smallest values are discarded.
- The largest and the smallest of the remaining values are averaged for the calculation of the midpoint value

Algorithm for clock correction value calculation (k=2).
Clock Correction

- The correction terms are used to adjust the number of microticks in each macrotick
Wake-Up Signal

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$dWU_{01}, dWU_{02}$</td>
<td>Duration of Data_0 phase in WU (*)</td>
<td>4</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>$dWU_{idle1}, dWU_{idle2}$</td>
<td>Duration of Idle phase in WU (*)</td>
<td>4</td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>$dWU$</td>
<td>Duration of valid wake up pattern</td>
<td>16</td>
<td>48</td>
<td>µs</td>
</tr>
</tbody>
</table>

(*) as seen by receiving BD.
Wakeup of two Channels in a Fault-Tolerant way by Coldstart Nodes

Node A startup node

Node B startup node

Node C Channel B

Channel A

Channel B
Startup

• Wakeup procedure has to be completed before *startup* can commence. The startup is performed on all channels synchronously.

• The action of initiating a startup process is called a *coldstart*. Only a limited number of nodes may initiate a startup, they are called the *coldstart nodes*. 
Startup

- When a coldstart node enters startup, it listens to its attached channels and attempts to receive FlexRay frames.
- If no communication is received, the node commences a coldstart attempt.
- A coldstart attempt begins with the transmission of a *collision avoidance symbol* (CAS).
- The CAS is succeeded by the first regular cycle *(Cycle 0: a startup frame)*.
- From Cycle 0 on, the node transmits its startup frame.
The Collision Avoidance Symbol (CAS) contains a long train of LOW bits.
Startup

- Since each coldstart node is allowed to perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path.

- This situation is resolved during the first four cycles after CAS transmission.

- As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it reenters the listen state.

- Consequently, only one node remains in this path.
Startup

• In Cycle 4, other coldstart nodes begin to transmit their startup frames.

• The node that initiated the coldstart collects all startup frames from Cycle 4 and 5 and performs clock correction.

• Then the node leaves startup and enters operation.
Path of the Integrating Coldstart Nodes

- When a coldstart node enters the startup, it listens to its attached channels and attempts to receive FlexRay frames.
- If communication is received, it tries to integrate to a transmitting coldstart node.
- It tries to receive a valid pair of startup frames.
- If these frame receptions have been successful, it performs clock correction in the following double-cycle.
Path of the Integrating Coldstart Nodes

• If clock correction does not signal any errors, it begins to transmit its startup frame; otherwise it reenters the listen state.

• If for the following three cycles the clock correction does not signal any errors and at least one other coldstart node is visible, the node leaves startup and enters operation.

• Thereby, it leaves startup at least one cycle after the node that initiated the coldstart.
Path of a Non-Coldstart Node

• When a non-coldstart node enters startup, it listens to its attached channels and tries to receive FlexRay frames.
• If communication is received, it tries to integrate to a transmitting coldstart node.
• It tries to receive a valid pair of startup frames to derive its schedule and clock correction.
Path of a Non-Coldstart Node

• In the following double-cycles, it tries to find at least two coldstart nodes that transmit startup frames.
• If this fails or if clock correction signals an error, the node aborts the integration attempt and tries anew.
Path of a Non-Coldstart Node

• After receiving valid startup frame pairs for two consecutive double-cycles from at least two coldstart nodes, the node leaves startup and enters operation.

• Thereby, it leaves startup at least two cycles after the node that initiated the coldstart.

• That means that all nodes of the cluster can leave startup at the end of cycle 7, just before entering cycle 8.
State Transitions for the Fault-Free Startup

Node A  
coldstart  
node  
POC  
state  
cycle  
schedule

Node B  
coldstart  
node
POC  
state

Node C  
POC  
state

Channel

Legend  
CAS  
CAS symbol  
S  
A: startup frame  
of node A  
S  
B: startup frame  
of node B  
C: frame of node C

earliest point in time for  
all nodes to leave startup

Flexray: A Time-Triggered  
Protocol  
© 2006 Syed Masud Mahmud
## Some System Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration of the channel idle delimiter.</td>
<td>11 bits</td>
</tr>
<tr>
<td>Maximum number of controllers in a cluster.</td>
<td>64</td>
</tr>
<tr>
<td>Maximum cycle counter value in a cluster.</td>
<td>63</td>
</tr>
<tr>
<td>Maximum cycle length.</td>
<td>16000 μs</td>
</tr>
<tr>
<td>Minimum number of microticks per macrotick during the offset correction phase.</td>
<td>20</td>
</tr>
<tr>
<td>Minimum number of microticks in a nominal (uncorrected) macrotick.</td>
<td>40</td>
</tr>
<tr>
<td>Highest slot ID number.</td>
<td>2047</td>
</tr>
<tr>
<td>Highest static slot ID number.</td>
<td>1023</td>
</tr>
<tr>
<td>Maximum propagation delay from the falling edge (in the BSS) in the transmit signal of node M to corresponding falling edge at the receiver of node N.</td>
<td>2.5 μs</td>
</tr>
</tbody>
</table>
## Some System Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum transmission phase in the dynamic segment due to the transmission limitation of an active star. (This value is equal to the longest configurable static slot of 2047(\mu)s).</td>
<td>2047 (\mu)s</td>
</tr>
<tr>
<td>Number of macroticks the action point is offset from the beginning of a static slots or symbol window.</td>
<td>1 – 31</td>
</tr>
<tr>
<td>Duration of the idle phase within a dynamic slot.</td>
<td>1 – 3 minislots</td>
</tr>
<tr>
<td>Duration of a minislot.</td>
<td>2 – 63 macroticks</td>
</tr>
<tr>
<td>Number of macroticks the minislot action point is offset from the beginning of a minislot.</td>
<td>1 – 31</td>
</tr>
<tr>
<td>Duration of a static slot</td>
<td>5 – 2047 macroticks</td>
</tr>
</tbody>
</table>
## Some System Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration of the symbol window.</td>
<td>0 – 87 macroticks</td>
</tr>
<tr>
<td>Number of macroticks in a communication cycle.</td>
<td>12 – 16000 macroticks</td>
</tr>
<tr>
<td>Number of minislots in the dynamic segment.</td>
<td>0 – 7994</td>
</tr>
<tr>
<td>Length of the Network Management vector in a cluster.</td>
<td>0 – 12</td>
</tr>
</tbody>
</table>
## Summary of TTCAN & FlexRay

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>TTCAN</th>
<th>FLEXRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Channels</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Configuration</td>
<td>bus</td>
<td>bus, hybrid, active star, cascaded star</td>
</tr>
<tr>
<td>Connection</td>
<td>twisted pair</td>
<td>twisted pair, fiber optic</td>
</tr>
<tr>
<td>Bus Length</td>
<td>40 m</td>
<td>depends on configuration</td>
</tr>
<tr>
<td>Bit Rate</td>
<td>1 Mbps</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>Data Length</td>
<td>0 - 8 bytes</td>
<td>0 - 254 bytes</td>
</tr>
<tr>
<td>Header Length</td>
<td>11 - 29 bits</td>
<td>40 bits</td>
</tr>
<tr>
<td>Error Detection</td>
<td>15-bit CRC</td>
<td>11-bit CRC &amp; 24-bit CRC</td>
</tr>
<tr>
<td>Hamming Distance</td>
<td>5</td>
<td>6 or 4</td>
</tr>
<tr>
<td>Feedback from Receiving Node</td>
<td>Error &amp; Acknowledge</td>
<td>Not in frame</td>
</tr>
</tbody>
</table>
THE END
References