

EXTENDS *Naturals, Sequences, Controller, Device*

A sequence of all variables

$vars \triangleq \langle mastershipVars, nodeVars, messageVars, streamVars, deviceVars \rangle$

The invariant asserts that the device will not allow a write from an older master if it has already accepted a write from a newer master. This is determined by comparing the *mastership* terms of accepted writes. For this invariant to hold, terms may only increase in the history of writes.

$TypeInvariant \triangleq$

$$\begin{aligned} &\wedge \forall x \in 1 \dots Len(history) : \\ &\quad \forall y \in x \dots Len(history) : \\ &\quad \quad \wedge history[x].term \leq history[y].term \\ &\quad \quad \wedge history[x].term = history[y].term \Rightarrow history[x].node = history[y].node \end{aligned}$$

$Init \triangleq$

$$\begin{aligned} &\wedge term = 0 \\ &\wedge master = Nil \\ &\wedge backups = \langle \rangle \\ &\wedge events = [n \in Nodes \mapsto \langle \rangle] \\ &\wedge mastership = [n \in Nodes \mapsto [term \mapsto 0, master \mapsto Nil, backups \mapsto \langle \rangle]] \\ &\wedge streamId = 0 \\ &\wedge sentTerm = [n \in Nodes \mapsto 0] \\ &\wedge isMaster = [n \in Nodes \mapsto FALSE] \\ &\wedge requestStream = [n \in Nodes \mapsto [id \mapsto 0, state \mapsto Closed]] \\ &\wedge requests = [n \in Nodes \mapsto \langle \rangle] \\ &\wedge responseStream = [n \in Nodes \mapsto [id \mapsto 0, state \mapsto Closed]] \\ &\wedge responses = [n \in Nodes \mapsto \langle \rangle] \\ &\wedge election = [n \in Nodes \mapsto 0] \\ &\wedge writeToken = 0 \\ &\wedge state = Stopped \\ &\wedge history = \langle \rangle \end{aligned}$$

$Next \triangleq$

$$\begin{aligned} &\vee \exists n \in Nodes : OpenStream(n) \\ &\quad \wedge UNCHANGED \langle deviceVars \rangle \\ &\vee \exists n \in Nodes : CloseStream(n) \\ &\quad \wedge UNCHANGED \langle deviceVars \rangle \\ &\vee \exists n \in Nodes : ConnectStream(n) \\ &\quad \wedge UNCHANGED \langle mastershipVars, nodeVars \rangle \\ &\vee \exists n \in Nodes : DisconnectStream(n) \\ &\quad \wedge UNCHANGED \langle mastershipVars, nodeVars \rangle \\ &\vee \exists n \in Nodes : JoinMastershipElection(n) \\ &\quad \wedge UNCHANGED \langle deviceVars \rangle \\ &\vee \exists n \in Nodes : LeaveMastershipElection(n) \\ &\quad \wedge UNCHANGED \langle deviceVars \rangle \end{aligned}$$

$$\begin{aligned}
& \vee \exists n \in \text{Nodes} : \text{LearnMastership}(n) \\
& \quad \wedge \text{UNCHANGED } \langle \text{device Vars} \rangle \\
& \vee \exists n \in \text{Nodes} : \text{SendMasterArbitrationUpdate}(n) \\
& \quad \wedge \text{UNCHANGED } \langle \text{device Vars} \rangle \\
& \vee \exists n \in \text{Nodes} : \text{HandleMasterArbitrationUpdate}(n) \\
& \quad \wedge \text{UNCHANGED } \langle \text{mastership Vars}, \text{node Vars} \rangle \\
& \vee \exists n \in \text{Nodes} : \text{ReceiveMasterArbitrationUpdate}(n) \\
& \quad \wedge \text{UNCHANGED } \langle \text{device Vars} \rangle \\
& \vee \exists n \in \text{Nodes} : \text{SendWriteRequest}(n) \\
& \quad \wedge \text{UNCHANGED } \langle \text{device Vars} \rangle \\
& \vee \exists n \in \text{Nodes} : \text{HandleWrite}(n) \\
& \quad \wedge \text{UNCHANGED } \langle \text{mastership Vars}, \text{node Vars} \rangle \\
& \vee \exists n \in \text{Nodes} : \text{ReceiveWriteResponse}(n) \\
& \quad \wedge \text{UNCHANGED } \langle \text{device Vars} \rangle \\
& \vee \text{Shutdown} \\
& \quad \wedge \text{UNCHANGED } \langle \text{mastership Vars}, \text{node Vars} \rangle \\
& \vee \text{Startup} \\
& \quad \wedge \text{UNCHANGED } \langle \text{mastership Vars}, \text{node Vars} \rangle
\end{aligned}$$

$$\text{Spec} \triangleq \text{Init} \wedge \Box[\text{Next}]_{\text{vars}}$$


---

\ \* Modification History  
\ \* Last modified *Mon Feb 25 16:25:03 PST 2019* by *jordanhalterman*  
\ \* Created *Thu Feb 14 11:33:03 PST 2019* by *jordanhalterman*