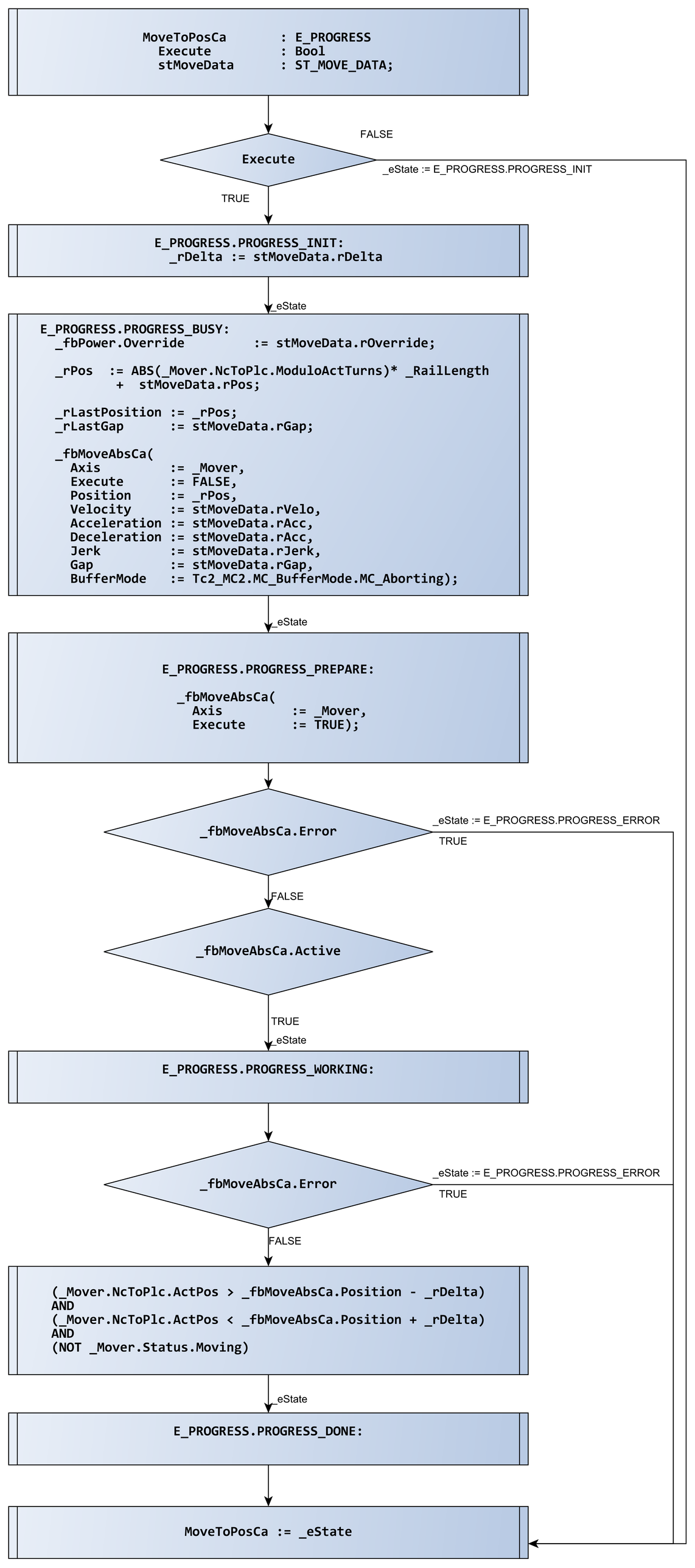
MoveToPosCa : E_PROGRESS
Execute : Bool
stMoveData : ST_MOVE_DATA;
Execute
FALSE
_eState := E_PROGRESS.PROGRESS_INIT
TRUE
E_PROGRESS.PROGRESS_INIT:
_rDelta := stMoveData.rDelta
_eState
E_PROGRESS.PROGRESS_BUSY:
_fbPower.Override := stMoveData.rOverride;
_rPos := ABS(_Mover.NcToPlc.ModuloActTurns)* _RailLength
+ stMoveData.rPos;
_rLastPosition := _rPos;
_rLastGap := stMoveData.rGap;
_fbMoveAbsCa(
Axis := _Mover,
Execute := FALSE,
Position := _rPos,
Velocity := stMoveData.rVelo,
Acceleration := stMoveData.rAcc,
Deceleration := stMoveData.rAcc,
Jerk := stMoveData.rJerk,
Gap := stMoveData.rGap,
BufferMode := Tc2_MC2.MC_BufferMode.MC_Aborting);
_eState
E_PROGRESS.PROGRESS_PREPARE:
_fbMoveAbsCa(
Axis := _Mover,
Execute := TRUE);
_fbMoveAbsCa.Error
_eState := E_PROGRESS.PROGRESS_ERROR
TRUE
FALSE
_fbMoveAbsCa.Active
TRUE
_eState
E_PROGRESS.PROGRESS_WORKING:
_fbMoveAbsCa.Error
_eState := E_PROGRESS.PROGRESS_ERROR
TRUE
FALSE
(_Mover.NcToPlc.ActPos > _fbMoveAbsCa.Position - _rDelta)
AND
(_Mover.NcToPlc.ActPos < _fbMoveAbsCa.Position + _rDelta)
AND
(NOT _Mover.Status.Moving)
_eState
E_PROGRESS.PROGRESS_DONE:
MoveToPosCa := _eState