```
MoveToPos        : E_PROGRESS
  Execute        : Bool
  stMoveData     : ST_MOVE_DATA;
```

```
Execute
```

```
E_PROGRESS.PROGRESS_INIT:
  _rDelta := stMoveData.rDelta
```

```
E_PROGRESS.PROGRESS_BUSY:
  _fbPower.Override         := stMoveData.rOverride;

  _rPos  := ABS(_Mover.NcToPlc.ModuloActTurns)* _RailLength
         +  stMoveData.rPos;

  _rLastPosition := _rPos;

  _fbMoveAbs(
    Axis         := _Mover,
    Execute      := FALSE,
    Position     := _rPos,
    Velocity     := stMoveData.rVelo,
    Acceleration := stMoveData.rAcc,
    Deceleration := stMoveData.rAcc,
    Jerk         := stMoveData.rJerk,
    BufferMode   := Tc2_MC2.MC_BufferMode.MC_Aborting);
```

```
E_PROGRESS.PROGRESS_PREPARE:

  _fbMoveAbs(
    Axis         := _Mover,
    Execute      := TRUE);
```

```
_fbMoveAbs.Error
```

```
_fbMoveAbs.Active
```

```
E_PROGRESS.PROGRESS_WORKING:
```

```
_fbMoveAbs.Error
```

```
(_Mover.NcToPlc.ActPos > _fbMoveAbs.Position - _rDelta)
AND
(_Mover.NcToPlc.ActPos < _fbMoveAbs.Position + _rDelta)
AND
(NOT _Mover.Status.Moving)
```

```
E_PROGRESS.PROGRESS_DONE:
```

```
MoveToPos := _eState
```

FALSE
_eState := E_PROGRESS.PROGRESS_INIT
TRUE
_eState
_eState
_eState := E_PROGRESS.PROGRESS_ERROR
TRUE
FALSE
TRUE
_eState
_eState := E_PROGRESS.PROGRESS_ERROR
TRUE
FALSE
_eState