

```

IF EnableACS
then

    (*Getting a smaller distance in camera 3 and 4, track 1. The train will stop around the middle of position 3 and position 4--*)

    if((TrainDataPrim.Position=15 or TrainDataPrim.Position=3 or TrainDataPrim.Position=4)and TrainDataPrim.Direction
        and TrainDataPrim.Track_Number=1) then

        if((TrainDataSek.Position=3 and not (TrainDataPrim.Position=4) and TrainDataSek.Direction
            and TrainDataSek.Track_Number=1) or (TrainDataThrd.Track_Number=1 and TrainDataThrd.Direction
            and TrainDataThrd.Position=3 and not (TrainDataPrim.Position=4)))
            then
                StopTrain:=true;
                Alarm_ACS_Trig := True;
                Alarm_ACS_Message := Stopped_because + Low_priority;

            elsif (TrainDataSek.Position=4 and TrainDataPrim.Pos_X_Cam3<1200 and TrainDataPrim.Position=3
                and TrainDataSek.Direction and TrainDataSek.Track_Number=1 )

                then
                    StopTrain:=true;
                    Alarm_ACS_Trig := True;
                    Alarm_ACS_Message := Stopped_because + Low_priority;

            elsif (TrainDataThrd.Position=4 and TrainDataPrim.Pos_X_Cam3<1200 and TrainDataPrim.Position=3
                and TrainDataThrd.Direction and TrainDataThrd.Track_Number=1)
                then
                    StopTrain:=true;
                    Alarm_ACS_Trig := True;
                    Alarm_ACS_Message := Stopped_because + Low_priority;

            elsif (((TrainDataPrim.Position=4)and(TrainDataPrim.Track_Number=1)) and (TrainDataSek.Position=18)
                and TrainDataPrim.Pos_X_Cam4<600)or(((TrainDataPrim.Position=4) and(TrainDataPrim.Track_Number=1))
                and ((TrainDataThrd.Position=18) and (TrainDataPrim.Pos_X_Cam4<600 and TrainDataPrim.Position=4)))) then
                StopTrain:=true;
                Alarm_ACS_Trig := True;
                Alarm_ACS_Message := Stopped_because + Low_priority;

            (*
            elsif(TrainDataPrim.Direction and TrainDataPrim.Position=15 and (((TrainDataSek.Track_Number=1)
                and TrainDataSek.Position=3) or (TrainDataThrd.Position=3 and TrainDataThrd.Track_Number=1))and TrainDataPrim.Track_Number=1)

                then
                    StopTrain:=true;
                    Alarm_ACS_Trig := True;
                    Alarm_ACS_Message := Stopped_because + Low_priority;
            *)

            else
                StopTrain:=false;
            end_if;

        (*-----*)

        (*Getting a smaller distance in camera 3 and 4, track 2. The train will stop around the middle of position 3 and position 4-----*)

        elsif((TrainDataPrim.Position=25 or TrainDataPrim.Position=3 or TrainDataPrim.Position=4)and TrainDataPrim.Direction
            and TrainDataPrim.Track_Number=2)
            then

            if((TrainDataSek.Position=3 and not (TrainDataPrim.Position=4) and TrainDataSek.Direction
                and TrainDataSek.Track_number=2) or (TrainDataThrd.Track_number=2 and TrainDataThrd.Direction
                and TrainDataThrd.Position=3 and not (TrainDataPrim.Position=4))) then
                StopTrain:=true;
                Alarm_ACS_Trig := True;
                Alarm_ACS_Message := Stopped_because + Low_priority;
            
```

```

elseif (TrainDataSek.Position=4 and TrainDataPrim.Pos_X_Cam3<1200 and TrainDataPrim.Position=3
and TrainDataSek.Direction and TrainDataSek.Track_number=2 )
then
StopTrain:=true;
Alarm_ACS_Trig := True;
Alarm_ACS_Message := Stopped_because + Low_priority;

elseif (TrainDataThrd.Position=4 and TrainDataPrim.Pos_X_Cam3<1200 and TrainDataPrim.Position=3
and TrainDataThrd.Direction and TrainDataThrd.Track_number=2)
then
StopTrain:=true;
Alarm_ACS_Trig := True;
Alarm_ACS_Message := Stopped_because + Low_priority;

elseif (((TrainDataPrim.Position=4)and(TrainDataPrim.Track_number=2)) and (TrainDataSek.Position=28)
and TrainDataPrim.Pos_X_Cam4<600)or(((TrainDataPrim.Position=4) and(TrainDataPrim.Track_number=2))
and ((TrainDataThrd.Position=28) and (TrainDataPrim.Pos_X_Cam4<600 and TrainDataPrim.Position=4))))
then
StopTrain:=true;
Alarm_ACS_Trig := True;
Alarm_ACS_Message := Stopped_because + Low_priority;

(*
elseif(TrainDataPrim.Direction and TrainDataPrim.Position=25 and (((TrainDataSek.Track_number=2) and TrainDataSek.Position=3)
or (TrainDataThrd.Position=3 and TrainDataThrd.Track_number=2))and TrainDataPrim.Track_number=2)
then
StopTrain:=true;
Alarm_ACS_Trig := True;
Alarm_ACS_Message := Stopped_because + Low_priority;
*)

else
StopTrain:=false;
end_if;

(*-----*)

(*When a train wants to shift track to tracknumber 2 in crossing 1, but is occupied-----*)

elseif TrainDataPrim.Position = 1 AND TrainDataPrim.Track_Number= 1 AND TrainDataSek.Position = 1 AND TrainDataSek.Track_Number = 2
AND Switch_Signals.Servo_1 AND Switch_Signals.Servo_3 AND TrainDataPrim.Priority > TrainDataSek.Priority
THEN
StopTrain:= True;
Alarm_ACS_Trig := True;
Alarm_ACS_Message := Stopped_because + Low_priority;
elseif TrainDataPrim.Position = 1 AND TrainDataPrim.Track_Number= 1 AND TrainDataThrd.Position = 1 AND TrainDataThrd.Track_Number = 2
AND Switch_Signals.Servo_1 AND Switch_Signals.Servo_3 AND TrainDataPrim.Priority > TrainDataThrd.Priority
THEN
StopTrain:= True;
Alarm_ACS_Trig := True;
Alarm_ACS_Message := Stopped_because + Low_priority;

(* Distance between CAM 3 and CAM 4*)
elseif TrainDataPrim.Position = 15 and TrainDataSek.Position = 3 AND TrainDataSek.Track_Number = 1 AND TrainDataSek.Pos_X_Cam3 > Cam3Pos2
THEN
StopTrain := True;
Alarm_ACS_Trig := True;
Alarm_ACS_Message := Stopped_because + Low_priority;
elseif TrainDataPrim.Position = 15 and TrainDataThrd.Position = 3 AND TrainDataThrd.Track_Number = 1 AND TrainDataThrd.Pos_X_Cam3 > Cam3Pos2
THEN
StopTrain := True;
Alarm_ACS_Trig := True;
Alarm_ACS_Message := Stopped_because + Low_priority;

(*When a train wants to shift track to tracknumber 1 in crossing 1, but is occupied*)
elseif TrainDataPrim.Position = 1 AND TrainDataPrim.Track_Number= 2 AND TrainDataSek.Position = 1 AND TrainDataSek.Track_Number = 1

```

```

    AND Switch_Signals.Servo_1 AND Switch_Signals.Servo_3 AND TrainDataPrim.Priority > TrainDataSek.Priority
  THEN
    StopTrain:= True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elsif TrainDataPrim.Position = 1 AND TrainDataPrim.Track_Number= 2 AND TrainDataThrd.Position = 1 AND TrainDataThrd.Track_Number = 1
  AND Switch_Signals.Servo_1 AND Switch_Signals.Servo_3 AND TrainDataPrim.Priority > TrainDataThrd.Priority
  THEN
    StopTrain:= True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

(*When a train wants to shift track to tracknumber 2 in crossing 2, but is occupied*)
elsif TrainDataPrim.Position = 2 AND TrainDataPrim.Track_Number= 1 AND TrainDataSek.Position = 2 AND TrainDataSek.Track_Number = 2
  AND Switch_Signals.Servo_5 AND Switch_Signals.Servo_6 AND TrainDataPrim.Priority > TrainDataSek.Priority
  THEN
    StopTrain:= True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elsif TrainDataPrim.Position = 2 AND TrainDataPrim.Track_Number= 1 AND TrainDataThrd.Position = 2 AND TrainDataThrd.Track_Number = 2
  AND Switch_Signals.Servo_5 AND Switch_Signals.Servo_6 AND TrainDataPrim.Priority > TrainDataThrd.Priority
  THEN
    StopTrain:= True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

(*When a train wants to shift track to tracknumber 1 in crossing 2, but is occupied*)
elsif TrainDataPrim.Position = 2 AND TrainDataPrim.Track_Number= 2 AND TrainDataSek.Position = 2 AND TrainDataSek.Track_Number = 1
  AND Switch_Signals.Servo_5 AND Switch_Signals.Servo_6 AND TrainDataPrim.Priority > TrainDataSek.Priority
  THEN
    StopTrain:= True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elsif TrainDataPrim.Position = 2 AND TrainDataPrim.Track_Number= 2 AND TrainDataThrd.Position = 2 AND TrainDataThrd.Track_Number = 1
  AND Switch_Signals.Servo_5 AND Switch_Signals.Servo_6 AND TrainDataPrim.Priority > TrainDataThrd.Priority
  THEN
    StopTrain:= True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

(*If the next position for the prim train equals the position for the second or third train, then the prim train stops*)
elsif TrainDataPrim.Next_Position = TrainDataSek.Position AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elsif TrainDataPrim.Next_Position = TrainDataThrd.Position AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

    (*When one train move forwards and the other moves backwards*)
elsif TrainDataPrim.Next_Position = TrainDataSek.Next_Position AND TrainDataPrim.Direction AND NOT TrainDataSek.Direction
  AND TrainDataPrim.Track_Number = TrainDataSek.Track_Number
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif TrainDataPrim.Next_Position = TrainDataThrd.Next_Position AND TrainDataPrim.Direction AND NOT TrainDataThrd.Direction
  AND TrainDataPrim.Track_Number = TrainDataThrd.Track_Number
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

```

```

elseif TrainDataPrim.Next_Position = TrainDataSek.Next_Position AND NOT TrainDataPrim.Direction AND TrainDataSek.Direction
    AND TrainDataPrim.Track_Number = TrainDataSek.Track_Number
THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elseif TrainDataPrim.Next_Position = TrainDataThrd.Next_Position AND NOT TrainDataPrim.Direction AND TrainDataThrd.Direction
    AND TrainDataPrim.Track_Number = TrainDataThrd.Track_Number
THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

(*CAM1 ANTI Collision*)

(*Forward: the train behind has higher speed than the other *)
elseif TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number) AND TrainDataPrim.Speed > TrainDataSek.Speed
    AND TrainDataPrim.Pos_X_Cam1 < TrainDataSek.Pos_X_Cam1
THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elseif TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number) AND TrainDataPrim.Speed > TrainDataThrd.Speed
    AND TrainDataPrim.Pos_X_Cam1 < TrainDataThrd.Pos_X_Cam1
THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

(*Backwards: the train forwards has higher speed than the train behinds *)
elseif NOT TrainDataPrim.Direction AND NOT TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number) AND TrainDataPrim.Speed > TrainDataSek.Speed
    AND TrainDataPrim.Pos_X_Cam1 > TrainDataSek.Pos_X_Cam1
THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elseif NOT TrainDataPrim.Direction AND NOT TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number) AND TrainDataPrim.Speed > TrainDataThrd.Speed
    AND TrainDataPrim.Pos_X_Cam1 > TrainDataThrd.Pos_X_Cam1
THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

(* One Train move forwards and the other backwards*)
elseif NOT TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number) AND TrainDataPrim.Pos_X_Cam1 > TrainDataSek.Pos_X_Cam1
THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elseif NOT TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number) AND TrainDataPrim.Pos_X_Cam1 < TrainDataSek.Pos_X_Cam1
THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

```

```

elseif NOT TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
  AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number) AND TrainDataPrim.Pos_X_Cam1 > TrainDataThrd.Pos_X_Cam1
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elseif NOT TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
  AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number) AND TrainDataPrim.Pos_X_Cam1 < TrainDataThrd.Pos_X_Cam1
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

    (*CAM2 ANTI Collision*)

    (*Forward: the train behind has higher speed than the other *)

elseif TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
  AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
  AND TrainDataPrim.Speed > TrainDataSek.Speed AND TrainDataPrim.Pos_X_Cam2 < TrainDataSek.Pos_X_Cam2
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elseif TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
  AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
  AND TrainDataPrim.Speed > TrainDataThrd.Speed AND TrainDataPrim.Pos_X_Cam2 < TrainDataThrd.Pos_X_Cam2
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

    (*Backwards: the train forwards has higher speed than the train behinds *)

elseif NOT TrainDataPrim.Direction AND NOT TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
  AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
  AND TrainDataPrim.Speed > TrainDataSek.Speed AND TrainDataPrim.Pos_X_Cam2 > TrainDataSek.Pos_X_Cam2
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elseif NOT TrainDataPrim.Direction AND NOT TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
  AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
  AND TrainDataPrim.Speed > TrainDataThrd.Speed AND TrainDataPrim.Pos_X_Cam2 > TrainDataThrd.Pos_X_Cam2
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

(* One Train move forwards and the other backwards-----*)

elseif NOT TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
  AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
  AND TrainDataPrim.Pos_X_Cam2 > TrainDataSek.Pos_X_Cam2
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elseif NOT TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
  AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
  AND TrainDataPrim.Pos_X_Cam2 < TrainDataSek.Pos_X_Cam2

```

```

    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif NOT TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
    AND TrainDataPrim.Pos_X_Cam2 > TrainDataThrd.Pos_X_Cam2
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif NOT TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
    AND TrainDataPrim.Pos_X_Cam2 < TrainDataThrd.Pos_X_Cam2
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

    (*CAM3 ANTI Collision*)

    (*Forward: the train behind has higher speed than the other-----*)

elsif TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
    AND TrainDataPrim.Speed > TrainDataSek.Speed AND TrainDataPrim.Pos_X_Cam3 > TrainDataSek.Pos_X_Cam3
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elsif TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
    AND TrainDataPrim.Speed > TrainDataThrd.Speed AND TrainDataPrim.Pos_X_Cam3 > TrainDataThrd.Pos_X_Cam3
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

    (*Backwards: the train forwards has higher speed than the train behinds *)

elsif NOT TrainDataPrim.Direction AND NOT TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
    AND TrainDataPrim.Speed > TrainDataSek.Speed AND TrainDataPrim.Pos_X_Cam3 < TrainDataSek.Pos_X_Cam3
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elsif NOT TrainDataPrim.Direction AND NOT TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
    AND TrainDataPrim.Speed > TrainDataThrd.Speed AND TrainDataPrim.Pos_X_Cam3 < TrainDataThrd.Pos_X_Cam3
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

    (* One Train move forwards and the other backwards-----*)

elsif NOT TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)

```

```

    AND TrainDataPrim.Pos_X_Cam3 > TrainDataSek.Pos_X_Cam3
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif NOT TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
    AND TrainDataPrim.Pos_X_Cam3 < TrainDataSek.Pos_X_Cam3
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif NOT TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
    AND TrainDataPrim.Pos_X_Cam3 > TrainDataThrd.Pos_X_Cam3
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif NOT TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
    AND TrainDataPrim.Pos_X_Cam3 < TrainDataThrd.Pos_X_Cam3
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

(*CAM4 ANTI Collision*)

(*Forward: the train behind has higher speed than the other-----*)

elsif TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
    AND TrainDataPrim.Speed > TrainDataSek.Speed AND TrainDataPrim.Pos_X_Cam4 > TrainDataSek.Pos_X_Cam4
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elsif TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
    AND TrainDataPrim.Speed > TrainDataThrd.Speed AND TrainDataPrim.Pos_X_Cam4 > TrainDataThrd.Pos_X_Cam4
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

(*Backwards: the train forwards has higher speed than the train behinds *)

elsif NOT TrainDataPrim.Direction AND NOT TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
    AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
    AND TrainDataPrim.Speed > TrainDataSek.Speed AND TrainDataPrim.Pos_X_Cam4 < TrainDataSek.Pos_X_Cam4
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

elsif NOT TrainDataPrim.Direction AND NOT TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
    AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
    AND TrainDataPrim.Speed > TrainDataThrd.Speed AND TrainDataPrim.Pos_X_Cam4 < TrainDataThrd.Pos_X_Cam4
    THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Low_priority;

```

```

(* One Train move forwards and the other backwards-----*)

elsif NOT TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
  AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
  AND TrainDataPrim.Pos_X_Cam4 > TrainDataSek.Pos_X_Cam4
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif NOT TrainDataPrim.Direction AND TrainDataSek.Direction AND TrainDataPrim.Position = TrainDataSek.Position
  AND (TrainDataPrim.Track_Number = TrainDataSek.Track_Number)
  AND TrainDataPrim.Pos_X_Cam4 < TrainDataSek.Pos_X_Cam4
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif NOT TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
  AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
  AND TrainDataPrim.Pos_X_Cam4 > TrainDataThrd.Pos_X_Cam4
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

elsif NOT TrainDataPrim.Direction AND TrainDataThrd.Direction AND TrainDataPrim.Position = TrainDataThrd.Position
  AND (TrainDataPrim.Track_Number = TrainDataThrd.Track_Number)
  AND TrainDataPrim.Pos_X_Cam4 < TrainDataThrd.Pos_X_Cam4
  THEN
    StopTrain := True;
    Alarm_ACS_Trig := True;
    Alarm_ACS_Message := Stopped_because + Direct_conflict;

    (*Parking-----*)
elsif TrainDataPrim.Position=31 and TrainDataPrim.Direction=true
  then
    StopTrain:= true;
    Alarm_ACS_Trig:=true;
    Alarm_ACS_Message:=TrainDataPrim.Name + Stopped_because + Parking + real_to_string( TrainDataPrim.Position);
elsif TrainDataPrim.Position=32 and TrainDataPrim.Direction=false
  then
    StopTrain:= true;
    Alarm_ACS_Trig:=true;
    Alarm_ACS_Message:=TrainDataPrim.Name + Stopped_because + Parking + real_to_string( TrainDataPrim.Position);
else
  StopTrain:= false;
  Alarm_ACS_Trig:=false;
END_IF;

else
  StopTrain:= false;
  Alarm_ACS_Trig:=false;
END_IF;

```