```
MODULE MainModule
    ! - - - - SUMMARY - - - - -
    ! CUSTOMER: xxxxxxx
    ! PROJECT: ROBOTIC WIPER BLADE VISION SYSTEM
    ! DESIGN BY: INDUSTRIAL AUTOMATED SYSTEMS INC.
    ! IAS JOB #: xxxxxx
    ! INSTALLATION DATE: 02/23/2017
    ! - - - - - - - - - - - - - -

    ! - - - - - - - - - - - - - -
    PROC main()
        !
        Reset doR2NotInHandoffArea;
        GoHome;
        !Stop;
        bPartDropped:=FALSE;
        bPermissionToEnter:=FALSE;
        bServosOut:=FALSE;
        bServosIn:=FALSE;
        !
        WHILE bPartDropped=FALSE AND bPermissionToEnter=FALSE DO
            !
            ! ---------- PREP FOR NEW CYCLE ----------
            IF bPartDropped=FALSE nNewCyclePrep_Start:=ClkRead(cycleTime);
            R01_NewCyclePrep;
            !
            ! ---------- servo error testing only ----------
            !MoveJ pInspectAppr2,v3000,z50,tBladeGripper;
            !MoveJ pInspectAppr1,v3000,z50,tBladeGripper;
            !ServoTestLoop:
            !MoveJ Offs(pFlippedFwdPick,0,0,10),v3000,z1,tBladeGripper;
            !MoveL Offs(pFlippedFwdPick,0,0,-42),v800,z1,tBladeGripper;
            !WaitRob\InPos;
            !rServosMoveIn;
            !rServosMoveOut;
            !GOTO ServoTestLoop;
            !
            ! ---------- check for dropped parts in handoff area ----------
            IF dinput(diHandoffPart1Present)=1 OR dinput(diHandoffPart2Present)=1 
THEN
                ErrWrite "DROPPED PART PRESENT IN HANDOFF AREA","Clear dropped part 
and start to continue";
                Stop;
            ENDIF
            !
            ! ---------- wait for parts ----------
            IF bPartDropped=FALSE nWaitForParts_Start:=ClkRead(cycleTime);
            R10_WaitForParts;
            Reset doR2NotInHandoffArea;
            !
            ! ---------- rotate first part ----------
            IF bPartDropped=FALSE nRotateFirstPart_Start:=ClkRead(cycleTime);
            R20_RotateFirstPart;
            !
            ! ---------- pick new parts ----------
            IF bPartDropped=FALSE nPickNewParts_Start:=ClkRead(cycleTime);
            R21_PickNewParts;
            !
            ! ---------- check for dropped parts ----------
            WaitRob\InPos;
            rDroppedPartsCheck;
            IF bPartDropped=TRUE THEN
                ErrWrite "PART DROPPED IN HANDOFF AREA DURING INITIAL PICK","Verify
```

```
that all areas are clear of parts before continuing";
                rRecover_Grippers;
                GOTO RejectBadParts;
            ENDIF
            !
            ! ---------- check for camera bypass ----------
            IF DOutput(SO_CameraBypass)=1 THEN
                MoveL Offs(pNewPartsPick,0,0,150),v3000,fine,tBladeGripper;
                MoveJ pInspectAppr1,v3000,z50,tBladeGripper;
                MoveJ pInspectAppr2,v4000,z50,tBladeGripper;
                MoveJ pStartPos,v4000,z50,tBladeGripper;
                WaitRob\InPos;
                GOTO DropGoodParts;
            ENDIF
            !
            ! ---------- flip parts forward ----------
            IF bPartDropped=FALSE nFlipPartsFwd_Start:=ClkRead(cycleTime);
            R22_FlipPartsFwd;
            !
            ! ---------- check for dropped parts ----------
            WaitRob\InPos;
            rDroppedPartsCheck;
            IF bPartDropped=TRUE THEN
                ErrWrite "PART DROPPED IN HANDOFF AREA DURING FIRST FLIP","Verify
that all areas are clear of parts before continuing";
                rRecover_Camera;
                GOTO RejectBadParts;
            ENDIF
            !
            ! ---------- inspect front side ----------
            IF bPartDropped=FALSE nInspectFront_Start:=ClkRead(cycleTime);
            R30_InspectFront;
            !
            ! ---------- flip parts reverse ----------
            IF bPartDropped=FALSE nFlipPartsRev_Start:=ClkRead(cycleTime);
            FlipPartsRev:
            R40_FlipPartsRev;
            !
            ! ---------- check for dropped parts ----------
            WaitRob\InPos;
            rDroppedPartsCheck;
            IF bPartDropped=TRUE THEN
                ErrWrite "PART DROPPED IN HANDOFF AREA DURING SECOND FLIP","Verify
that all areas are clear of parts before continuing";
                rRecover_Camera;
                GOTO RejectBadParts;
            ENDIF
            !
            ! ---------- inspect part backside ----------
            IF bPartDropped=FALSE nInspectBack_Start:=ClkRead(cycleTime);
            R50_InspectBackTop;
            R51_InspectBackBottom;
            !IF btoppartbad=TRUE AND bbottompartbad=TRUE GOTO RejectBadParts;
            !
            ! ---------- check for warp and mark top part ----------
            IF bPartDropped=FALSE nWarpAndMark_Start:=ClkRead(cycleTime);
            WarpAndMarkTopPart:
            R60_WarpAndMarkTopPart;
            !
            ! ---------- check for dropped parts ----------
            WaitRob\InPos;
            rDroppedPartsCheck;
            IF bPartDropped=TRUE THEN
```

```
                TPErase;
                ErrWrite "PART DROPPED IN LASER MARKER AREA","Verify that all areas 
are clear of parts before continuing";
                rRecover_Laser;
                GOTO RejectBadParts;
            ENDIF
            !
            ! ---------- check for warp and mark bottom part ----------
            R61_WarpAndMarkBottomPart;
            !
            ! ---------- check for dropped parts ----------
            WaitRob\InPos;
            rDroppedPartsCheck;
            IF bPartDropped=TRUE THEN
                TPErase;
                ErrWrite "PART DROPPED IN LASER MARKER AREA","Verify that all areas 
are clear of parts before continuing";
                GOTO RejectBadParts;
            ENDIF
            !
            ! ---------- reject bad parts ----------
            IF bPartDropped=FALSE nDropParts_Start:=ClkRead(cycleTime);
            RejectBadParts:
            IF bTopPartBad=FALSE AND bBottomPartBad=FALSE GOTO DropGoodParts;
            R70_RejectBadParts;
            !
            ! ---------- drop good parts ----------
            DropGoodParts:
            !IF bTopPart=FALSE AND bBottomPart=FALSE GOTO EndOfCycle;
            IF bTopPartBad=TRUE AND bBottomPartBad=TRUE GOTO EndOfCycle;
            R80_DropGoodParts;
            !
            ! ----- display cycle summary -----
            EndOfCycle:
            rCycleSummary;
        ENDWHILE

    ENDPROC

    !---------- END OF MAIN ROUTINE ----------
    !----------------------------------------
    !
    !---------- BEGIN SUB ROUTINES ----------
    !---------------------------------------
    PROC R01_NewCyclePrep()
        ! initialize the system by setting necessary values and outputs to begin a 
new cycle
        ! move out, open, and rotate (Counter Clockwise) grippers
        Reset SO_CycleFinished;
        !
        ! ----- gripper servos -----
        rGrippersOpen;
        rGrippersRotateCouClk;
        IF bServosOut=FALSE THEN
            rServosMoveOut;
        ENDIF
        rServoFaultCheck;
        !
        ! ----- laser marker -----
        Reset doLaserStartMark;
        Reset SO_PartBad;
        Reset doLaserUserIn2;
        rLaserMarkJobChange;
        !
        ! ----- inspection camera -----
        SetDO doCameraBottomBlade,0;
        SetDO doCameraPartBackSide,0;
        Reset doCameraTrigger;
        nCameraTopPartPassScore:=0;
        nCameraBottomPartPassScore:=0;
        ! check that camera is online
        IF DOutput(So_CameraBypass)=0 THEN
            IF bCamOnLine=FALSE THEN
                ErrWrite "ISSUE WITH CAMERA OFF LINE","Check vision system to 
resolve. Parts will bypass until camera is online";
                Stop;
            ENDIF
        ENDIF
        !
        ! ----- inspection robot -----
        Reset doEOATVac1;
        Reset doEOATVac2;
        bTopPartBad:=FALSE;
        bBottomPartBad:=FALSE;
        b1stPtPresent:=FALSE;
        b2ndPtPresent:=FALSE;
        bTopPart:=FALSE;
        bBottomPart:=FALSE;
        bPartGood:=FALSE;
        bPartBad:=FALSE;
        bTopPartWarped:=FALSE;
        bBottomPartWarped:=FALSE;
        bPartDropped:=FALSE;
        !
        ! ----- inspection robot and mold robot comms -----
        Reset doR2NotInHandoffArea;
        ! check for permission to enter
        !IF bPermissionToEnter=TRUE THEN
        !    MoveJ pHome,v500,fine,tBladeGripper;
        !    bPermissionToEnter:=FALSE;
        !    TPWrite "ROBOT IS IN HOME POSITION RESTART WHEN READY";
        !    Stop;
        !ENDIF
        SetDO SO_SystemRunning,1;
    ENDPROC

    PROC R10_WaitForParts()
        ! check handoff area presence sensors for dropped parts
        HandoffAreaDroppedPartsRecheck1:
        IF dinput(diHandoffPart1Present)=1 OR dinput(diHandoffPart2Present)=1 THEN
            ErrWrite "DROPPED PART PRESENT IN HANDOFF AREA","Clear dropped part and 
start to continue";
            Stop;
            GOTO HandoffAreaDroppedPartsRecheck1;
        ENDIF
        ! notify mold robot that inspection robot is ready to accept parts
        Set doR2NotInHandoffArea;
        ! rotate gripper 180 degrees in order to rotate first part
        MoveJ pRotatePartPickWaitAppr1,v3000,z50,tBladeGripper;
        MoveJ pRotatePartPickWaitAppr2,v3000,z50,tBladeGripper;
        ! move to wait position
        MoveJ pRotatePartPickWait,v3000,z50,tBladeGripper;
        ! wait for mold robot to signal that it has entered handoff area
        WaitDI diR1NotInHandoffArea,0;
        Set doHandoffBlowoff1;
        Set doHandoffBlowoff2;
        ! wait for handoff parts to be sensed
        WaitUntil dinput(diHandoffPart1Present)=1 OR 
dinput(diHandoffPart2Present)=1;
        rHandoffVacOn;
        ! wait for mold robot to signal it has left handoff area then pulse blowoff 
to realign parts if misguided
        WaitDI diR1NotInHandoffArea,1;
        rR1HandoffStatusCheck;
        rHandoffVacOff;
        ! verify that both parts are present
        HandoffAreaDroppedPartsRecheck2:
        IF dinput(diHandoffPart1Present)=0 OR dinput(diHandoffPart2Present)=0 THEN
            WaitTime 0.5;
            IF dinput(diHandoffPart1Present)=0 OR dinput(diHandoffPart2Present)=0 
THEN
                ErrWrite "PART MISSING IN HANDOFF AREA","If 1 part present then pp 
to main and clear.";
                Stop;
                GOTO HandoffAreaDroppedPartsRecheck2;
            ENDIF
        ENDIF
        !
    ENDPROC

    PROC R20_RotateFirstPart()
        !
        ! ----- pick first part -----
        ! notify mold robot that inspection robot is in handoff area
        Reset doR2NotInHandoffArea;
        MoveJ Offs(pRotatePartPick,0,0,100),v3000,z1,tBladeGripper;
        !turn top EOAT vac on
        Set doEOATVac1;
        !move to final pick pos
        MoveL pRotatePartPick,v1000,fine,tBladeGripper;
        !turn handoff area vac off for first part only
        WaitRob\InPos;
        SetDO doHandoffVac1,0;
        PulseDO\PLength:=0.1,doHandoffBlowoff1;
        !pull off by 3mm and wait for vac confirm (added to help with dropping of 
parts)
        MoveL Offs(pRotatePartPick,0,0,3),v30,z1,tBladeGripper;
        WaitDI diEOATVac1,1\MaxTime:=0.25\TimeFlag:=bTimeOut;
        ! pull off 100mm
        MoveL Offs(pRotatePartPick,0,0,100),v1000,z1,tBladeGripper;
        !
        ! ----- rotate first part -----
        ! move away to allow roatation
        MoveJ pRotatePart1,v3000,z10,tBladeGripper;
        MoveJ pRotatePart2,v3000,z10,tBladeGripper;
        ! move to place rotated part
        !
        ! ----- place rotated part -----
        MoveJ Offs(pNewPartsPick,0,0,100),v3000,z1,tBladeGripper;
        MoveL pNewPartsJustify1,v600,fine,tBladeGripper;
        MoveL pNewPartsJustify2,v600,fine,tBladeGripper;
        MoveL pNewPartsJustify3,v600,fine,tBladeGripper;
        MoveL pNewPartsJustify1,v600,fine,tBladeGripper;
        !MoveL Offs(pNewPartsPick,0,0,-3),v600,z1,tBladeGripper;
        ! place rotated part
        rEOATVacOff;
        rHandoffVacOn;
        MoveL Offs(pNewPartsPick,0,0,20),v600,z1,tBladeGripper;
        WaitRob\InPos;
        ! hold rotated part with handoff vac
    ENDPROC

    PROC R21_PickNewParts()
        !
        ! ----- pick both parts -----
        MoveL Offs(pNewPartsPick,0,0,20),v600,z1,tBladeGripper;
        rEOATVacOn;
        MoveL pNewPartsPick,v600,fine,tBladeGripper;
        WaitRob\InPos;
        rHandoffVacOff;
        !pull off by 3mm and wait for vac confirm (added to help with dropping of
parts)
        MoveL Offs(pNewPartsPick,0,0,nPickPullup),v30,z1,tBladeGripper;
        WaitUntil DInput(diEOATVac1)=1 AND
DInput(diEOATVac2)=1\MaxTime:=0.25\TimeFlag:=bTimeOut;
        MoveL Offs(pNewPartsPick,0,0,40),v600,fine,tBladeGripper;
    ENDPROC

    PROC R22_FlipPartsFwd()
        rGrippersOpen;
        !
        ! ----- place parts in grippers -----
        !MoveL pPreFirstPick,v3000,z5,tBladeGripper;
        MoveL pFlippedRevPick,v800,fine,tBladeGripper;
        WaitRob\ZeroSpeed;
        rServosMoveIn;
        rGrippersClose;
        rEOATVacOff;
        !
        ! ----- pull robot out and flip parts -----
        MoveL Offs(pFlippedFwdPick,0,0,50),v3000,z1,tBladeGripper;
        WaitRob\InPos;
        rGrippersRotateClk;
        !
        ! ----- pick flipped parts -----
        rEOATVacOn;
        MoveL Offs(pFlippedFwdPick,0,0,-3),v800,fine,tBladeGripper;
        WaitRob\ZeroSpeed;
        WaitTime 0.25;
        !pull off by 3mm and wait for vac confirm (added to help with dropping of
parts)
        !MoveL Offs(pFlippedFwdPick,0,0,nPickPullup),v30,z1,tBladeGripper;
        !WaitRob\ZeroSpeed;
        WaitUntil DInput(diEOATVac1)=1 AND
DInput(diEOATVac2)=1\MaxTime:=0.25\TimeFlag:=bTimeOut;
        IF bTimeOut=TRUE THEN
            bTimeOut:=FALSE;
            !rEOATVacOff;
            WaitTime 0.5;
            MoveL Offs(pFlippedFwdPick,0,-2,-3),v800,fine,tBladeGripper;
            WaitRob\ZeroSpeed;
            !rEOATVacOn;
            WaitTime 0.5;
            !MoveL Offs(pFlippedFwdPickRedo,0,0,15),v800,fine,tBladeGripper;
            !MoveL Offs(pFlippedFwdPickRedo,0,0,-4),v800,fine,tBladeGripper;
        ENDIF
        rGrippersOpen;
        !rServosMoveOut;
        !
        ! ----- pull robot out and move to inspection -----
        MoveL Offs(pFlippedFwdPick,0,60,0),v800,z1,tBladeGripper;
        MoveL Offs(pFlippedFwdPick,0,60,50),v800,z1,tBladeGripper;
        MoveJ pInspectAppr1,v3000,z50,tBladeGripper;
        MoveJ pInspectAppr2,v4000,z50,tBladeGripper;
        !
    ENDPROC

    PROC R30_InspectFront()
        MoveJ p1stCameraPos,v600,fine,tBladeGripper;
        Reset doCameraPartBackSide;
        Reset doCameraBottomBlade;
        WaitTime 0.1;
        PulseDO\PLength:=0.1,doCameraReset;
        WaitTime 0.1;
        rInspectTrigger;
        ! pos 1
        MoveJ p1stCameraPos2,v600,fine,tBladeGripper;
        rInspectionResults;
        ! pos 1
        rInspectTrigger;
        MoveJ p1stCameraPos3,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p1stCameraPos4,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p1stCameraPos5,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p1stCameraPos6,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p1stCameraPos7,v600,fine,tBladeGripper;
        rInspectionResults;
        WaitTime 0.25;
        rInspectTrigger;
        ! pos 7
        rInspectionResults;
        ! pos 7
        !
        ! ---------- BOTTOM PART INSPECT ----------
        MoveJ p1stCameraPos8,v600,fine,tBladeGripper;
        WaitTime 0.1;
        Reset doCameraPartBackSide;
        Set doCameraBottomBlade;
        WaitTime 0.1;
        PulseDO\PLength:=0.1,doCameraReset;
        WaitTime 0.25;
        rInspectTrigger;
        ! pos 8
        MoveJ p1stCameraPos9,v600,fine,tBladeGripper;
        rInspectionResults;
        ! pos 8
        rInspectTrigger;
        MoveJ p1stCameraPos10,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p1stCameraPos11,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p1stCameraPos12,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p1stCameraPos13,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p1stCameraPos14,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        !pos 14
        rInspectionResults;
        ! pos 14
        !
    ENDPROC

    PROC R40_FlipPartsRev()
        !
        ! ----- prep -----
        rR1HandoffStatusCheck;
        rGrippersOpen;
        !
        ! ----- move to gripper area -----
        MoveJ pInspectAppr2,v3000,z50,tBladeGripper;
        MoveJ pInspectAppr1,v3000,z50,tBladeGripper;
        !
        ! ----- place both parts -----
        MoveJ Offs(pFlippedFwdPick,0,60,50),v3000,z1,tBladeGripper;
        MoveL Offs(pFlippedFwdPick,0,60,0),v800,z1,tBladeGripper;
        MoveL pFlippedFwdPick,v600,fine,tBladeGripper;
        WaitRob\ZeroSpeed;
        !rServosMoveIn;
        rGrippersClose;
        rEOATVacOff;
        MoveL Offs(pFlippedRevPick,0,0,50),v3000,z1,tBladeGripper;
        !
        ! ----- flip parts -----
        WaitRob\InPos;
        rGrippersRotateCouClk;
        !
        ! ----- pick flipped parts -----
        rEOATVacOn;
        MoveL Offs(pFlippedRevPick,0,0,-3),v800,fine,tBladeGripper;
        WaitRob\ZeroSpeed;
        WaitTime 0.25;
        !pull off by 3mm and wait for vac confirm (added to help with dropping of 
parts)
        !MoveL Offs(pFlippedRevPick,0,0,nPickPullup),v30,z1,tBladeGripper;
        WaitUntil DInput(diEOATVac1)=1 AND 
DInput(diEOATVac2)=1\MaxTime:=0.25\TimeFlag:=bTimeOut;
        rGrippersOpen;

        !
        ! ----- pull robot out and move to inspection -----
        MoveL Offs(pFlippedRevPick,0,60,0),v800,z1,tBladeGripper;
        MoveL Offs(pFlippedRevPick,0,60,50),v3000,z1,tBladeGripper;
        rServosMoveOut;
        MoveJ pInspectAppr1,v3000,z50,tBladeGripper;
        MoveJ pInspectAppr2,v4000,z50,tBladeGripper;
        !
    ENDPROC

    PROC R50_InspectBackTop()
        !move to position
        MoveJ p2ndCameraPos15,v600,fine,tBladeGripper;
        WaitTime 0.1;
        Set doCameraPartBackSide;
        Reset doCameraBottomBlade;
        WaitTime 0.1;
        PulseDO\PLength:=0.1,doCameraReset;
        WaitTime 0.1;
        ! trigger camera
        rInspectTrigger;
        ! pos 15
        MoveJ p2ndCameraPos16,v600,fine,tBladeGripper;
        rInspectionResults;
        ! pos 15
        rInspectTrigger;
        MoveJ p2ndCameraPos17,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p2ndCameraPos18,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p2ndCameraPos19,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p2ndCameraPos20,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveJ p2ndCameraPos21,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        ! pos 21
        rInspectionResults;
        ! pos 21
    ENDPROC

    PROC R51_InspectBackBottom()
        MoveJ p2ndCameraPos22,v3000,fine,tBladeGripper;
        WaitTime 0.1;
        Set doCameraPartBackSide;
        Set doCameraBottomBlade;
        WaitTime 0.1;
        PulseDO\PLength:=0.1,doCameraReset;
        WaitTime 0.1;
        rInspectTrigger;
        ! pos 22
        MoveL p2ndCameraPos23,v600,fine,tBladeGripper;
        rInspectionResults;
        ! pos 22
        rInspectTrigger;
        MoveL p2ndCameraPos24,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveL p2ndCameraPos25,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveL p2ndCameraPos26,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveL p2ndCameraPos27,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        MoveL p2ndCameraPos28,v600,fine,tBladeGripper;
        rInspectionResults;
        rInspectTrigger;
        ! pos 28
        rInspectionResults;
        WaitTime 0.1;
        ! pos 28
        !
```

```
        ! ----- set good or bad -----
        IF nCameraTopPartPassScore<14 THEN
            bTopPartBad:=TRUE;
        ENDIF
        IF nCameraBottomPartPassScore<14 THEN
            bBottomPartBad:=TRUE;
        ENDIF
        TPErase;
        TPWrite "Top Part Pass Score: "\Num:=nCameraTopPartPassScore;
        TPWrite "Bottom Part Pass Score: "\Num:=nCameraBottomPartPassScore;
        !
        ! ----- pull out of inspect area -----
        MoveJ pInspectAppr2,v4000,z10,tBladeGripper;
    ENDPROC

    PROC R60_WarpAndMarkTopPart()
        !
        ! ----- write results to laser marker -----
        ! this is so that you dont have to wait for the job to change if parts are 
good
        IF bTopPartBad=TRUE Set SO_PartBad;
        IF bTopPartBad=FALSE Reset SO_PartBad;
        WaitTime 0.1;
        rLaserMarkJobChange;
        !
        ! ---------- top part place ----------
        MoveJ pInspectAppr2,v4000,z10,tBladeGripper;
        MoveJ pWarpTopPlaceAppr1,v4000,z20,tBladeGripper;
        MoveJ pWarpTopPlaceAppr2,v3000,z10,tBladeGripper;
        MoveL Offs(pWarpTopPlace,0,0,25),v1000,fine,tBladeGripper;
        MoveL pWarpTopPlace,v1000,fine,tBladeGripper;
        SetDO doEOATVac1,0;
        PulseDO\PLength:=0.1,doEOATBlowoff1;
        !
        ! ----- pull out from laser marker area to read warp -----
        MoveL Offs(pWarpTopPlace,0,0,25),v1000,fine,tBladeGripper;
        MoveJ pWarpTopPlaceAppr2,v1000,z5,tBladeGripper;
        WaitRob\InPos;
        IF DInput(diWarpLeftBad)=1 OR DInput(diWarpRightBad)=1 THEN
            bTopPartBad:=TRUE;
            bTopPartWarped:=TRUE;
        ENDIF
        !
        ! ----- trigger laser marker -----
        IF DInput(diLaserPoweredOn)=1 THEN
            ! if part bad does not match laser job then change
            IF bTopPartBad=TRUE AND DOutput(SO_PartBad)=0 THEN
                Set SO_PartBad;
                rLaserMarkJobChange;
                WaitTime 0.75;
            ENDIF
            WaitDO doLaserJobSelect,0;
            rLaserMarkExecute;
            WaitUntil 
DInput(diLaserMarkingInProgress)=1\MaxTime:=5.0\TimeFlag:=bTimeOut;
            IF bTimeOut=true THEN
                ErrWrite "LASER MARKER DID NOT INITIALIZE","Parts Will be Rejected";
                bTimeOut:=FALSE;
                bTopPartBad:=TRUE;
            ENDIF
        ELSEIF DOutput(SO_TS_BypassLaserMark)=1 THEN
            TPWrite "Laser Marker Bypassed";
        ENDIF
        IF DInput(diLaserPoweredOn)=0 THEN
            ErrWrite "LASER MARKER NOT POWERED ON","Parts Will be Rejected Until 
Power is Restored";
            bTopPartBad:=TRUE;
        ENDIF
        !
        ! ---------- top part pick ---------
        MoveJ pWarpTopPlaceAppr2,v1000,z10,tBladeGripper;
        MoveL Offs(pWarpTopPlace,0,0,25),v1000,fine,tBladeGripper;
        MoveL pWarpTopPlace,v1000,fine,tBladeGripper;
        ! wait for laser marker to finish
        WaitUntil DInput(diLaserMarkingInProgress)=0\MaxTime:=60\TimeFlag:=bTimeOut;
        IF bTimeOut=true THEN
            bTimeOut:=FALSE;
            ErrWrite "LASER MARKER TIMED OUT","Parts will be rejected until problem 
is resolved";
            bTopPartBad:=TRUE;
        ENDIF
        !
        rEOATVacOn;
        MoveL Offs(pWarpTopPlace,0,0,25),v1000,fine,tBladeGripper;
    ENDPROC

    PROC R61_WarpAndMarkBottomPart()
        !
        ! ----- write results to laser marker -----
        ! this is so that you dont have to wait for the job to change if parts are 
good
        IF bBottomPartBad=TRUE Set SO_PartBad;
        IF bBottomPartBad=FALSE Reset SO_PartBad;
        WaitTime 0.1;
        rLaserMarkJobChange;
        !
        ! ---------- bottom part place ----------
        MoveL Offs(pPlaceBackWarp,0,0,15),v1000,fine,tBladeGripper;
        MoveL pPlaceBackWarp,v1000,fine,tBladeGripper;
        SetDO doEOATVac2,0;
        PulseDO\PLength:=0.1,doEOATBlowoff2;
        MoveL Offs(pPlaceBackWarp,0,0,25),v1000,fine,tBladeGripper;
        !
        ! ----- evaluate for warp after pulling out -----
        MoveL pBackWarpPullout,v1000,z1,tBladeGripper;
        MoveL pWarpTopPlaceAppr2,v1000,z1,tBladeGripper;
        WaitRob\InPos;
        ! evaluate for warp
        IF DInput(diWarpLeftBad)=1 OR DInput(diWarpRightBad)=1 THEN
            bBottomPartBad:=TRUE;
            bBottomPartWarped:=TRUE;
        ENDIF
        !
        ! ----- trigger laser marker after warp -----
        ! so that vac does not affect warp reading
        !
        IF DInput(diLaserPoweredOn)=1 THEN
            ! if part bad does not match laser job then change laser job
            IF bBottomPartBad=TRUE AND DOutput(SO_PartBad)=0 THEN
                Set SO_PartBad;
                rLaserMarkJobChange;
                WaitTime 0.75;
            ENDIF
            WaitDO doLaserJobSelect,0;
            rLaserMarkExecute;
            WaitUntil
```

```
DInput(diLaserMarkingInProgress)=1\MaxTime:=5.0\TimeFlag:=bTimeOut;
            IF bTimeOut=true THEN
                ErrWrite "LASER MARKER DID NOT INITIALIZE","Parts Will be Rejected";
                bTimeOut:=FALSE;
                bBottomPartBad:=TRUE;
            ENDIF
        ELSEIF DOutput(SO_TS_BypassLaserMark)=1 THEN
            TPWrite "Laser Marker Bypassed";
        ENDIF
        IF DInput(diLaserPoweredOn)=0 THEN
            ErrWrite "LASER MARKER NOT POWERED ON","Parts Will be Rejected Until 
Power is Restored";
            bBottomPartBad:=TRUE;
        ENDIF
        ! ---------- BOTTOM PART PICK ----------
        MoveL pWarpTopPlaceAppr2,v1000,z1,tBladeGripper;
        MoveJ pBackWarpPullout,v1000,z5,tBladeGripper;
        MoveL Offs(pPlaceBackWarp,0,0,15),v1000,fine,tBladeGripper;
        MoveL Offs(pPlaceBackWarp,0,0,-12),v1000,fine,tBladeGripper;
        rEOATVacOn;
        !make sure laser mark of bottom part has completed
        WaitUntil DInput(diLaserMarkingInProgress)=0\MaxTime:=60\TimeFlag:=bTimeOut;
        IF bTimeOut=true THEN
            bTimeOut:=FALSE;
            ErrWrite "LASER MARKER TIMED OUT","Parts will be rejected until problem 
is resolved";
            bBottomPartBad:=TRUE;
        ENDIF
        !pull out of laser marker area
        MoveL Offs(pPlaceBackWarp,0,0,25),v1000,fine,tBladeGripper;
        MoveJ pBackWarpPullout,v1000,z5,tBladeGripper;
        MoveJ pWarpTopPlaceAppr2,v1000,z10,tBladeGripper;
        MoveJ pWarpTopPlaceAppr1,v3000,z20,tBladeGripper;
        MoveJ pStartPos,v3000,z20,tBladeGripper;
    ENDPROC

    PROC R70_RejectBadParts()
        MoveJ pStartPos,v3000,z10,tBladeGripper;
        MoveJ pPreBadPartsDrop,v3000,z20,tBladeGripper;
        MoveJ pBadPartsDrop,v3000,z10,tBladeGripper;
        WaitRob\InPos;
        IF bTopPartBad=TRUE THEN
            SetDO doEOATVac1,0;
            PulseDO\PLength:=0.5,doEOATBlowoff1;
            bTopPart:=FALSE;
            Incr nTotalBadParts;
            WaitDI diEOATVac1,0;
        ENDIF
        IF bBottomPartBad=TRUE THEN
            SetDO doEOATVac2,0;
            PulseDO\PLength:=0.5,doEOATBlowoff2;
            bBottomPart:=FALSE;
            Incr nTotalBadParts;
            WaitDI diEOATVac2,0;
        ENDIF
        MoveJ pPreBadPartsDrop,v3000,z20,tBladeGripper;
        MoveJ pHome,v3000,fine,tBladeGripper;
    ENDPROC

    PROC R80_DropGoodParts()
        !
        ! ----- move into good drop pos -----
        MoveJ pPreGoodPartsDrop,v3000,z20,tBladeGripper;
        MoveJ pGoodPartsDrop,v3000,z5,tBladeGripper;
        WaitRob\InPos;
        !
        ! ----- drop parts and adjust counts -----
        rEOATVacOff;
        IF bTopPart=TRUE THEN
            bTopPart:=FALSE;
            Incr nTotalGoodParts;
        ENDIF
        IF bBottomPart=FALSE THEN
            bBottomPart:=FALSE;
            Incr nTotalGoodParts;
        ENDIF
        !
        ! ----- move back to start pos -----
        MoveJ pPreGoodPartsDrop,v3000,z20,tBladeGripper;
        MoveJ pStartPos,v3000,fine,tBladeGripper;
    ENDPROC

    PROC rHandoffVacOn()
        Reset doHandoffBlowoff1;
        Reset doHandoffBlowoff2;
        Set doHandoffVac1;
        Set doHandoffVac2;
        !WaitTime 0.1;
    ENDPROC

    PROC rHandoffVacOff()
        ! first slot
        SetDO doHandoffVac1,0;
        PulseDO\PLength:=0.1,doHandoffBlowoff1;
        ! second slot
        SetDO doHandoffVac2,0;
        PulseDO\PLength:=0.1,doHandoffBlowoff2;
    ENDPROC

    PROC rR1HandoffStatusCheck()
        IF DInput(diR1NotInHandoffArea)=1 THEN
            Reset SO_SystemWaiting;
            RETURN ;
        ELSE
            !TPWrite " * Handoff Area Occupied by Big Robot *";
            Set SO_SystemWaiting;
            WaitUntil diR1NotInHandoffArea=1;
            Reset SO_SystemWaiting;
            !TPErase;
        ENDIF
    ENDPROC

    PROC rEOATVacOn()
        Set doEOATVac1;
        Set doEOATVac2;
        WaitTime 0.25;
    ENDPROC

    PROC rEOATVacOff()
        SetDO doEOATVac1,0;
        PulseDO\PLength:=0.1,doEOATBlowoff1;
        SetDO doEOATVac2,0;
        PulseDO\PLength:=0.1,doEOATBlowoff2;
        WaitTime 0.1;
    ENDPROC
    PROC rDroppedPartsCheck()
        ! make sure that vac is on
        Set doEOATVac1;
        Set doEOATVac2;
        WaitTime 0.1;
        !check for vac confirmation signal
        IF diEOATVac1=0 OR diEOATVac2=0 THEN
            ! notify that part has been dropped
            bPartDropped:=TRUE;
            ! set both parts to bad so that they are rejected
            bTopPartBad:=TRUE;
            bBottomPartBad:=TRUE;
            ! increase dropped part counts
            IF diEOATVac1=0 THEN
                btoppart:=FALSE;
                Incr ntotaldroppedpartstop;
            ENDIF
            IF diEOATVac2=0 THEN
                bbottompart:=FALSE;
                Incr ntotaldroppedpartsbottom;
            ENDIF
        ENDIF
    ENDPROC

    PROC rHandoffPartCheck()
        Set doHandoffVac1;
        Set doHandoffVac2;
        b1stPtPresent:=TRUE;
        b2ndPtPresent:=TRUE;
        WaitTime 0.1;
        IF diHandoffVac1=0 THEN
            Reset doHandoffVac1;
            b1stPtPresent:=FALSE;
        ENDIF
        IF diHandoffVac2=0 THEN
            Reset doHandoffVac2;
            b2ndPtPresent:=FALSE;
        ENDIF
    ENDPROC

    PROC rCycleSummary()
        !
        TPErase;
        TPWrite "----- CYCLE SUMMARY -----";
        rCycleTimeDisplay_1;
        IF DOutput(SO_DisplayCycleTimeBreakdown)=1 THEN
            !TPWrite " - New Cycle Start Time: "\Num:=nNewCyclePrep_Start;
            TPWrite " - Wait for Parts Start Time: "\Num:=nWaitForParts_Start;
            TPWrite " - Rotate First Part Start Time: "\Num:=nRotateFirstPart_Start;
            TPWrite " - Pick New Parts Start Time: "\Num:=nPickNewParts_Start;
            TPWrite " - Flip Parts Fwd Start Time: "\Num:=nFlipPartsFwd_Start;
            TPWrite " - Inspect Front Start Time: "\Num:=nInspectFront_Start;
            TPWrite " - Flip Parts Rev Start Time: "\Num:=nFlipPartsRev_Start;
            TPWrite " - Inspect Back Start Time: "\Num:=nInspectBack_Start;
            TPWrite " - Warp and Mark Start Time: "\Num:=nWarpAndMark_Start;
            TPWrite " - Drop Parts Start Time: "\Num:=nDropParts_Start;
        ELSE
            !
            ! ----- top part -----
            TPWrite "----- PART RESULTS -----";
            IF bTopPartBad=FALSE AND bTopPartWarped=FALSE TPWrite " TOP - PASSED";
            IF bTopPartBad=FALSE AND bTopPartWarped=TRUE TPWrite " TOP - FAILED 
(Warp check only)";
```

```
            IF bTopPartBad=TRUE AND bTopPartWarped=FALSE TPWrite " TOP - FAILED
(Inspection only)";
            IF bTopPartBad=TRUE AND bTopPartWarped=TRUE TPWrite " TOP - FAILED
(Insp. & Warp check)";
            !
            ! ----- bottom part -----
            IF bBottomPartBad=FALSE AND bBottomPartWarped=FALSE TPWrite " BOTTOM -
PASSED";
            IF bBottomPartBad=FALSE AND bBottomPartWarped=TRUE TPWrite " BOTTOM -
FAILED (Warp check only)";
            IF bBottomPartBad=TRUE AND bBottomPartWarped=FALSE TPWrite " BOTTOM -
FAILED (Inspection only)";
            IF bBottomPartBad=TRUE AND bBottomPartWarped=TRUE TPWrite " BOTTOM -
FAILED (Insp. & Warp check)";
            !
            ! ----- counts -----
            TPWrite "----- COUTNERS -----";
            TPWrite "Good Parts = "\Num:=nTotalGoodParts;
            TPWrite "Bad Parts = "\Num:=nTotalBadParts;
            !TPWrite "Warped Parts = "\Num:=nTotalWarpedParts;
            TPWrite "Dropped Parts Top = "\Num:=nTotalDroppedPartsTop;
            TPWrite "Dropped Parts Bottom = "\Num:=nTotalDroppedPartsBottom;
        ENDIF
    ENDPROC

    PROC rCycleTimeDisplay_1()
        !Reads and calculates last cycle time, and average since last start
        !TPErase;
        ClkStop cycleTime;
        CycleTimeRead:=ClkRead(cycleTime);
        TPWrite "----- TIMERS -----";
        TPWrite "Time for this cycle = "\Num:=CycleTimeRead;
        IF AverageCycleTime=0 AverageCycleTime:=40;
        AverageCycleTime:=(AverageCycleTime+CycleTimeRead)/2;
        TPWrite "Current Average Cycle Time = "\Num:=AverageCycleTime;
        !PartsPerShift:=(28800/AverageCycleTime)*2;
        !TPWrite "Approximate parts per shift with average cycle time =
"\Num:=PartsPerShift;
        !hour:=GetTime(\Hour);
        !minute:=GetTime(\min);
        !WaitTime 0.25;
        !Calculates the current shift and saves last shift's production based on
time
        !IF hour = 5 AND (minute >= 45 AND minute <=46) THEN
        !    ThirdShiftParts;
        !ELSEIF hour = 13 AND (minute >= 45 AND minute <=46) THEN
        !    FirstShiftParts;
        !ELSEIF hour=21 AND (minute >= 45 AND minute <=46) THEN
        !    SecondShiftParts;
        !ENDIF
        ClkReset cycleTime;
        ClkStart cycleTime;
    ENDPROC

    PROC rZDummy_CameraMoveSpeedTest()
        MoveL p1stCameraPos,v600,fine,tBladeGripper;
        ClkStart cycleTime;
        WaitRob\InPos;
        WaitRob\ZeroSpeed;
        MoveL p1stCameraPos2,v600,fine,tBladeGripper;
        WaitRob\InPos;
        WaitRob\ZeroSpeed;
        MoveL p1stCameraPos3,v600,fine,tBladeGripper;
        WaitRob\InPos;
        WaitRob\ZeroSpeed;
        MoveL p1stCameraPos4,v600,fine,tBladeGripper;
        WaitRob\InPos;
        WaitRob\ZeroSpeed;
        MoveL p1stCameraPos5,v600,fine,tBladeGripper;
        WaitRob\InPos;
        WaitRob\ZeroSpeed;
        MoveL p1stCameraPos6,v600,fine,tBladeGripper;
        WaitRob\InPos;
        WaitRob\ZeroSpeed;
        MoveL p1stCameraPos7,v600,fine,tBladeGripper;
        WaitRob\InPos;
        WaitRob\ZeroSpeed;
        ClkStop cycleTime;
        CycleTimeRead:=ClkRead(cycleTime);
        ClkReset cycleTime;
        TPWrite "Inspection Time: "\Num:=CycleTimeRead;
        MoveJ p1stCameraPos2,v600,fine,tBladeGripper;
    ENDPROC

    !---------- END SUB ROUTINES ----------
    !--------------------------------------

ENDMODULE
```