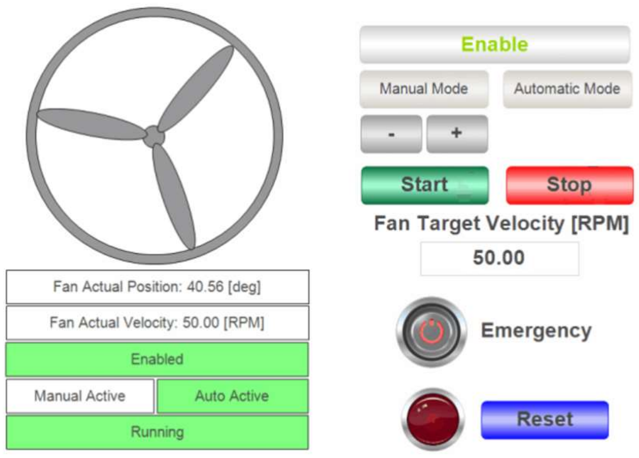
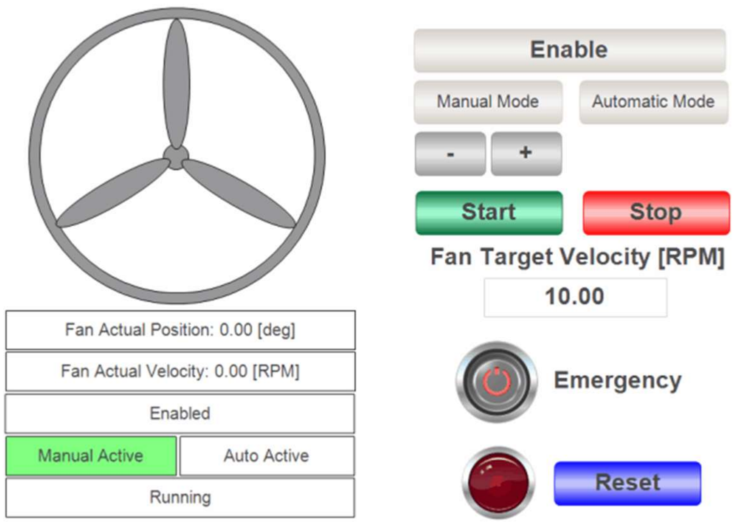
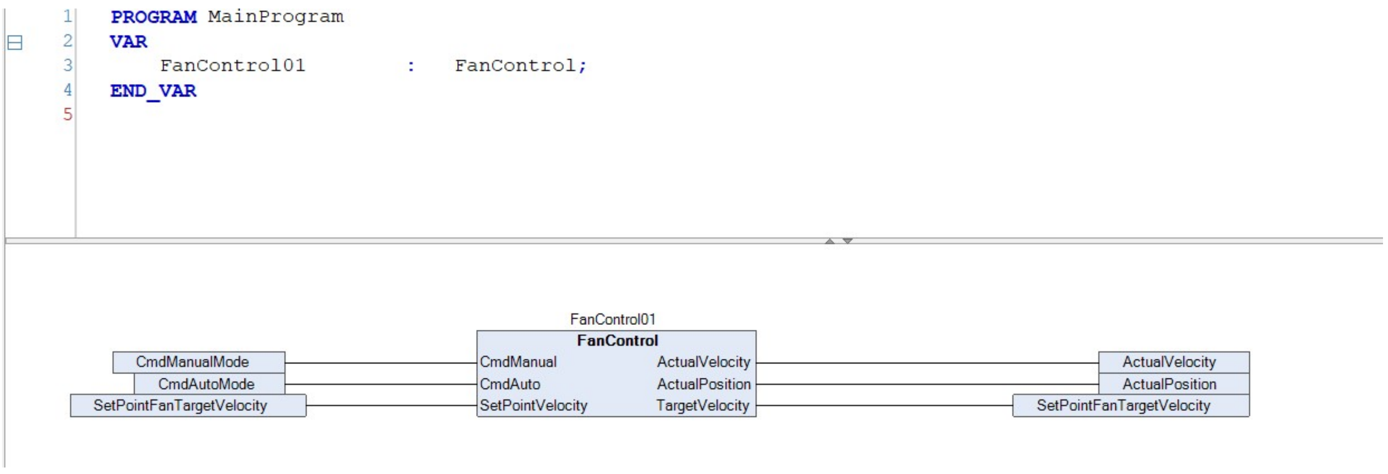
PLCopen – Fan Control / Continous Motion

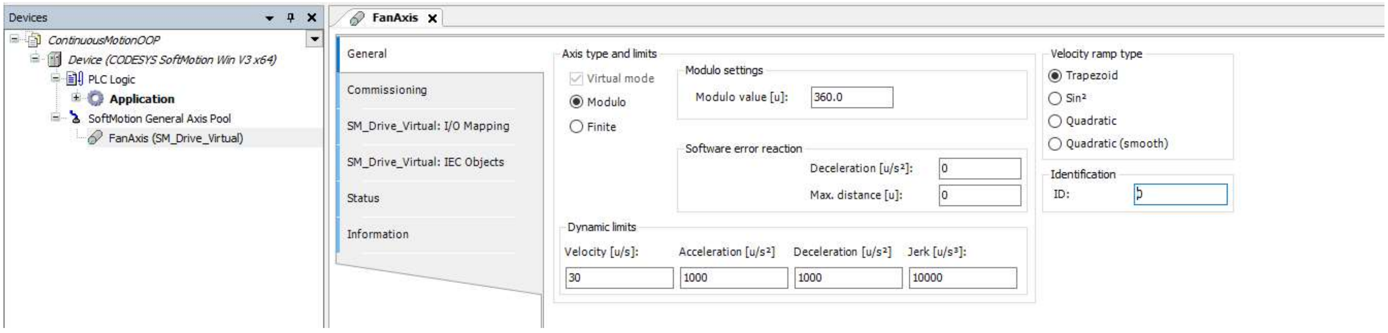
Simple project for fan velocity control with PLCopen motion function blocks.







```
PROGRAM MainProgram
VAR
    FanControl01        :    FanControl;
END_VAR
```


Devices
ContinuousMotionOOP
Device (CODESYS SoftMotion Win V3 x64)
PLC Logic
Application
SoftMotion General Axis Pool
FanAxis (SM_Drive_Virtual)
FanAxis
General
Commissioning
SM_Drive_Virtual: I/O Mapping
SM_Drive_Virtual: IEC Objects
Status
Information

Axis type and limits

- [✓] Virtual mode
- [● Modulo]
- ○ Finite

Modulo settings

Modulo value [u]: 360.0

Software error reaction

Deceleration [u/s²]: 0

Max. distance [u]: 0

Dynamic limits

Velocity [u/s]:	Acceleration [u/s²]	Deceleration [u/s²]	Jerk [u/s³]:
30	1000	1000	10000

Velocity ramp type

- [● Trapezoid]
- ○ Sin²
- ○ Quadratic
- ○ Quadratic (smooth)

Identification

ID: 0

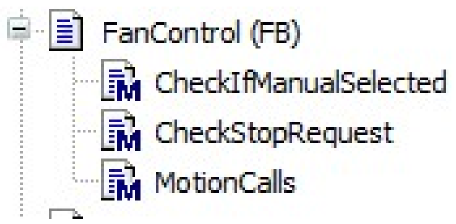
```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE ET_MachineMode :
(
      Manual := 0,
      Auto := 1
);
END_TYPE
```

```
VAR_GLOBAL
      // HMI Commands
      CmdEnable                           :     BOOL;
      CmdManualMode                       :     BOOL;
      CmdAutoMode                         :     BOOL;
      CmdJogFwd                           :     BOOL;
      CmdJogBwd                           :     BOOL;
      CmdStart                            :     BOOL;
      CmdStop                             :     BOOL;
      CmdReset                            :     BOOL;
      CmdEmergencyButton                  :     BOOL := TRUE;
      SetPointFanTargetVelocity           :     LREAL;
      CmdAlarmLed                         :     BOOL;

      // HMI Leds - Feedbacks
      StsEnabled                          :     BOOL;
      StsMachineMode                      :     ET_MachineMode;
      StsRunning                          :     BOOL;

      ActualPosition                      :     LREAL;
      ActualVelocity                      :     LREAL;

      // Global Parameters
      MinVelocity                         :     LREAL  := 10.0;
      MaxVelocity                         :     LREAL  := 100.0;
      Acceleration                        :     LREAL  := 300.0;
      Deceleration                        :     LREAL  := 300.0;
      EmergencyDeceleration               :     LREAL  := 500.0;
END_VAR
```



```
FUNCTION_BLOCK FanControl
VAR_INPUT
       CmdManual                 :      BOOL;
       CmdAuto                   :      BOOL;
       SetPointVelocity          :      LREAL;
END_VAR
VAR_OUTPUT
       ActualVelocity            :      LREAL;
       ActualPosition            :      LREAL;
       TargetVelocity            :      LREAL;
END_VAR
VAR

       // state machine
       SeqState                  :      (DISABLED,
                                        WAIT_FOR_POWER_ON,
                                        MANUAL_MODE,
                                        AUTO_MODE_WAIT_FOR_START,
                                        AUTO_MODE_START_MOVE_VELOCITY,
                                        AUTO_MODE_WAIT_FOR_IN_VELOCITY,
                                        AUTO_MODE_IN_VELOCITY,
                                        AUTO_MODE_STOPPING,
                                        SELECT_AUTO_MANUAL,
                                        EMERGENCY_STOPPING,
                                        EMERGENCY_DISABLING,
                                        EMERGENCY_WAIT_FOR_RESET,
                                        EMERGENCY_RESET_DONE);



       // MC FBs
       fbPowerFan                       :      MC_Power;
       fbResetFan                       :      MC_Reset;
       fbReadActualPositionFan          :      MC_ReadActualPosition;
       fbReadActualVelocityFan          :      MC_ReadActualVelocity;
       fbReadAxisError                  :      MC_ReadAxisError;
       fbJogFan                         :      MC_Jog;
       fbMoveVelocityFan                :      MC_MoveVelocity;
       fbStopFan                        :      MC_Stop;
END_VAR

-----------------------------------------------------------------------------------------------

// Call Motion FBs for axis
MotionCalls();


// handle manual - auto pushbuttons
IF CmdManual THEN StsMachineMode := ET_MachineMode.Manual; END_IF;
IF CmdAuto THEN StsMachineMode := ET_MachineMode.Auto; END_IF;

// handle target velocity limitation
TargetVelocity := LIMIT(MinVelocity, SetPointVelocity, MaxVelocity);

// *** state machine ***
CASE SeqState OF

       DISABLED:
       //  *** machine disable state ***

                     StsEnabled           := FALSE;
                     StsRunning           := FALSE;
```

```
            IF CmdEnable AND CmdEmergencyButton THEN
                  fbPowerFan.bDriveStart :=  TRUE;
                  fbPowerFan.bRegulatorOn := TRUE;
                  SeqState := WAIT_FOR_POWER_ON;
            END_IF


WAIT_FOR_POWER_ON:
// *** wait for power on ***

      IF fbPowerFan.Status THEN
            StsEnabled := TRUE;
            IF StsMachineMode = ET_MachineMode.Manual THEN
                  SeqState := MANUAL_MODE;
            ELSIF StsMachineMode = ET_MachineMode.Auto THEN
                  SeqState := AUTO_MODE_WAIT_FOR_START;
            END_IF;
      END_IF

MANUAL_MODE:
// *** manual mode ***

      // convert RPM to deg/sec.
      fbJogFan.Velocity := TargetVelocity * 6.0;
      fbJogFan.JogForward := CmdJogFwd;
      fbJogFan.JogBackward := CmdJogBwd;

      // transition to auto mode
      IF StsMachineMode = ET_MachineMode.Auto THEN
            fbJogFan.JogForward := FALSE;
            fbJogFan.JogBackward := FALSE;
            fbStopFan.Execute := TRUE;
            fbStopFan.Deceleration := Deceleration;
            SeqState := SELECT_AUTO_MANUAL;
      END_IF


AUTO_MODE_WAIT_FOR_START:
// *** auto mode  - wait for start ***

      IF CmdStart THEN
            SeqState := AUTO_MODE_START_MOVE_VELOCITY;
            StsRunning := TRUE;
      END_IF

      // transition to manual mode
      IF StsMachineMode = ET_MachineMode.Manual THEN
            fbMoveVelocityFan.Execute := FALSE;
            fbStopFan.Execute := TRUE;
            fbStopFan.Deceleration := Deceleration;
            SeqState := SELECT_AUTO_MANUAL;
      END_IF

AUTO_MODE_START_MOVE_VELOCITY:
// *** auto mode - start move velocity ***

      // convert RPM to deg/sec.
      fbMoveVelocityFan.Velocity := TargetVelocity * 6.0;
      fbMoveVelocityFan.Execute := TRUE;
      IF fbMoveVelocityFan.Busy THEN
            SeqState := AUTO_MODE_WAIT_FOR_IN_VELOCITY;
      END_IF

      // check if manual mode selected
      CheckIfManualSelected();

      // stop request
      CheckStopRequest();

AUTO_MODE_WAIT_FOR_IN_VELOCITY:
// *** auto mode - wait for InVelocity ***

      fbMoveVelocityFan.Execute := FALSE;
```

```
            // target velocity changed -> needs execute again (state
            // AUTO_MODE_START_MOVE_VELOCITY)
            // convert RPM to deg/sec.
            IF fbMoveVelocityFan.Velocity <> TargetVelocity * 6.0 THEN
                    SeqState := AUTO_MODE_START_MOVE_VELOCITY;
            END_IF

            IF fbMoveVelocityFan.InVelocity THEN
                    SeqState := AUTO_MODE_IN_VELOCITY;
            END_IF

            // check if manual mode selected
            CheckIfManualSelected();

            // stop request
            CheckStopRequest();

    AUTO_MODE_IN_VELOCITY:
    // *** auto mode - velocity reached ***

            // target velocity chnaged -> needs execute again (state
AUTO_MODE_START_MOVE_VELOCITY)
            // convert RPM to deg/sec.
            IF fbMoveVelocityFan.Velocity <> TargetVelocity * 6.0 THEN
                    SeqState := AUTO_MODE_START_MOVE_VELOCITY;
            END_IF

            // check if manual mode selected
            CheckIfManualSelected();

            // stop request
            CheckStopRequest();


    AUTO_MODE_STOPPING:
    // *** auto mode - stopping state ***

            IF fbStopFan.Done THEN
                    fbStopFan.Execute := FALSE;
                    SeqState := AUTO_MODE_WAIT_FOR_START;
            END_IF


    SELECT_AUTO_MANUAL:
    // *** transition:  manual -> auto or auto -> manual ***

            IF fbStopFan.Done THEN
                    fbStopFan.Execute := FALSE;
                    IF StsMachineMode = ET_Machinemode.Auto THEN
                            SeqState := AUTO_MODE_WAIT_FOR_START;
                    ELSIF StsMachineMode = ET_Machinemode.Manual THEN
                            SeqState := MANUAL_MODE;
                    END_IF
            END_IF

    EMERGENCY_STOPPING:
    // *** emergency - stopping ***

            IF fbStopFan.Done THEN
                    fbStopFan.Execute := FALSE;
                    StsRunning := FALSE;
                    SeqState := EMERGENCY_DISABLING;
            END_IF

    EMERGENCY_DISABLING:
    // *** emergency - disabling ***

            fbPowerFan.bDriveStart := FALSE;
            fbPowerFan.bRegulatorOn := FALSE;

            IF NOT fbPowerFan.Status THEN
                    StsEnabled := FALSE;
                    SeqState := EMERGENCY_WAIT_FOR_RESET;
            END_IF
```

```
		EMERGENCY_WAIT_FOR_RESET:
		// *** emergency - wait for reset ***

			IF CmdReset THEN
				IF fbReadAxisError.Error THEN
					fbResetFan.Execute := TRUE;
					SeqState := EMERGENCY_RESET_DONE;
				ELSE
					CmdAlarmLed := FALSE;
					SeqState := DISABLED;
				END_IF
			END_IF

		EMERGENCY_RESET_DONE:
		// *** emergency - wait for reset done ***

			IF fbResetFan.Done THEN
				CmdAlarmLed := FALSE;
				SeqState := DISABLED;
			END_IF

	END_CASE


	// disable transition
	IF NOT CmdEnable THEN
		fbPowerFan.bDriveStart := FALSE;
		fbPowerFan.bRegulatorOn := FALSE;
		fbMoveVelocityFan.Execute := FALSE;
		fbJogFan.JogForward := FALSE;
		fbJogFan.JogBackward := FALSE;
		fbstopFan.Execute := FALSE;
		SeqState := DISABLED;
	END_IF

	// emergency/error transition
	IF (NOT CmdEmergencyButton OR fbReadAxisError.Error) AND SeqState > DISABLED AND SeqState <
	EMERGENCY_STOPPING THEN
		fbStopFan.Execute := TRUE;
		fbMoveVelocityFan.Execute := FALSE;
		fbJogFan.JogForward := FALSE;
		fbJogFan.JogBackward := FALSE;
		fbStopFan.Deceleration := EmergencyDeceleration;
		CmdAlarmLed := TRUE;
		SeqState := EMERGENCY_STOPPING;
	END_IF

	// update actual position and actual velocity
	IF fbReadActualPositionFan.Valid THEN
		ActualPosition := fbReadActualPositionFan.Position;
	END_IF

	IF fbReadActualVelocityFan.Valid THEN
		// convert deg/sec. to RPM
		ActualVelocity := fbReadActualVelocityFan.Velocity / 6.0;
	END_IF

	----------------------------------------------------------------------------------------------

	METHOD CheckIfManualSelected : BOOL
	VAR_INPUT
	END_VAR

	// This method checks if manual mode was selcted and if yes sets respectively commands

	IF StsMachineMode = ET_MachineMode.Manual THEN
		StsRunning := FALSE;
		fbMoveVelocityFan.Execute := FALSE;
		fbStopFan.Execute := TRUE;
		fbStopFan.Deceleration := Deceleration;
		SeqState := SELECT_AUTO_MANUAL;
	END_IF
```

```
METHOD CheckStopRequest
VAR_INPUT
END_VAR

// check stop request
IF CmdStop THEN
       StsRunning := FALSE;
       fbMoveVelocityFan.Execute := FALSE;
       fbStopFan.Execute := TRUE;
       fbStopFan.Deceleration := Deceleration;
       SeqState :=  AUTO_MODE_STOPPING;
END_IF
```

```
METHOD MotionCalls
VAR_INPUT
END_VAR

// This method calls all MC FBs

fbPowerFan(
       Axis:= FanAxis,
       Enable:= TRUE,
       bRegulatorOn:= ,
       bDriveStart:= ,
       Status=> ,
       bRegulatorRealState=> ,
       bDriveStartRealState=> ,
       Busy=> ,
       Error=> ,
       ErrorID=> );


fbResetFan(
       Axis:= FanAxis,
       Execute:= ,
       Done=> ,
       Busy=> ,
       Error=> ,
       ErrorID=> );


fbReadActualPositionFan(
       Axis:= FanAxis,
       Enable:= TRUE,
       Valid=> ,
       Busy=> ,
       Error=> ,
       ErrorID=> ,
       Position=> );


fbReadActualVelocityFan(
       Axis:= FanAxis,
       Enable:= TRUE,
       Valid=> ,
       Busy=> ,
       Error=> ,
       ErrorID=> ,
       Velocity=> );


fbReadAxisError(
       Axis:= FanAxis,
       Enable:= TRUE,
       Valid=> ,
       Busy=> ,
       Error=> ,
       ErrorID=> ,
       AxisError=> ,
       AxisErrorID=> ,
       SWEndSwitchActive=> );
```

```
fbJogFan(
	Axis:= FanAxis,
	JogForward:= ,
	JogBackward:= ,
	Velocity:= ,
	Acceleration:= Acceleration,
	Deceleration:= Deceleration,
	Jerk:= ,
	Busy=> ,
	CommandAborted=> ,
	Error=> ,
	ErrorId=> );


fbMoveVelocityFan(
	Axis:= FanAxis,
	Execute:= ,
	Velocity:= ,
	Acceleration:= Acceleration,
	Deceleration:= Deceleration,
	Jerk:= ,
	Direction:= Mc_DIRECTION.positive,
	BufferMode:= ,
	InVelocity=> ,
	Busy=> ,
	Active=> ,
	CommandAborted=> ,
	Error=> ,
	ErrorID=> );


fbStopFan(
	Axis:= FanAxis,
	Execute:= ,
	Deceleration:= ,
	Jerk:= ,
	Done=> ,
	Busy=> ,
	Error=> ,
	ErrorID=> );
```