

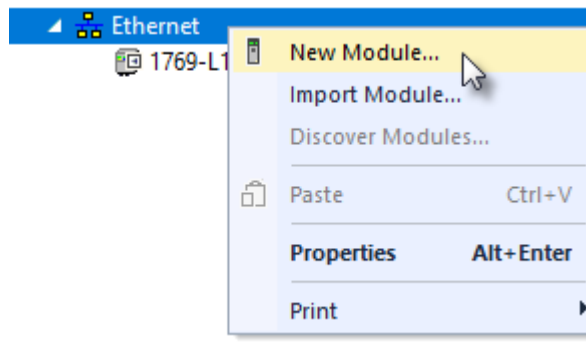
INTEGRATION-READY EXAMPLE USE INSTRUCTIONS

DESCRIPTION

The Integration-Ready Example was designed for integration into most existing machine control programs while providing expandable and customizable functionality. If you're new to ClearPath-IP, there's also a Lightweight Example showing a straightforward introduction to servo control using the provided AOIs.

STUDIO 5000 SETUP

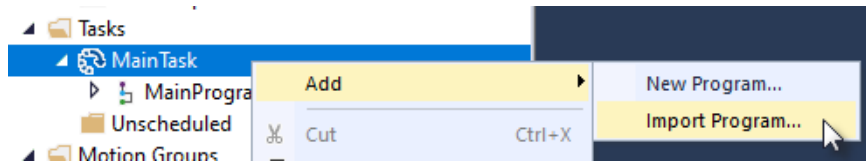
1. If you haven't already, start by registering the appropriate EDS file using Studio 5000's *EDS Hardware Installation Tool*. There are separate EDS files for each I/O HUB variant (IO-HUB-2-R, IO-HUB-4-R, and IO-HUB-4-E).
2. Create a new EtherNet/IP module in your project:



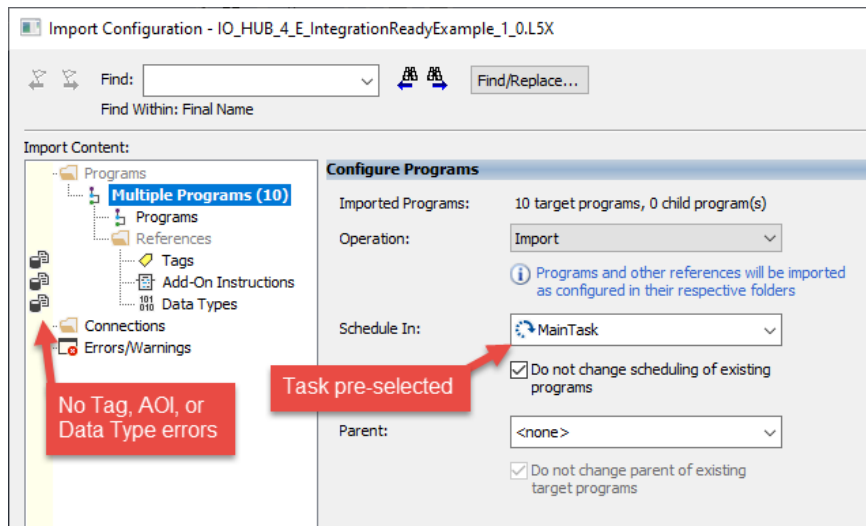
Filter by "Teknic" and select the I/O HUB that matches your hub part number. Click "Create".

3. In the *Module Properties* window, name your hub module and enter its IP Address (use the ClearView 3.0 setup program to get or set the hub's IP Address).
In the "Connection" tab, the RPI can be set as low as 1 ms for fastest communication between the PLC and I/O HUB.

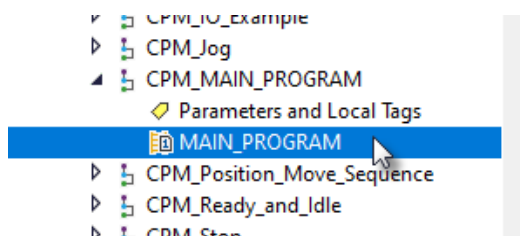
4. In your project, right click on the task you wish to use for motion, then click “Add” > “Import Program...”. Select the Integration Ready .L5X file associated with your hub. This will import all Programs, Tags, and AOIs used by the example.



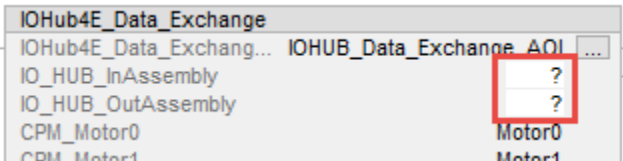
Note: the Import Configuration pop-up window should pre-select the option to import all target programs and schedule them in your task. Ensure the Import Configuration tool does not detect errors. If errors are present, check that you’ve selected the correct hub EDS file and the .L5X file associated with that module part number.



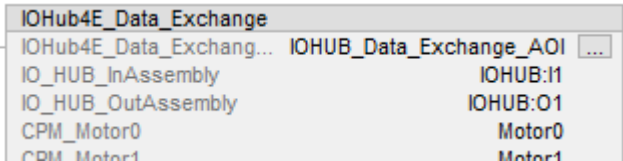
5. Open the CPM_MAIN_PROGRAM routine.



The first rung of this routine contains the Data Exchange AOI. This rung will be in an error condition because the IO_HUB_InAssembly and IO_HUB_OutAssembly parameters are missing their arguments.



Enter the assembly tags associated with the I/O HUB module created earlier. The module in this case was called “IOHUB”.



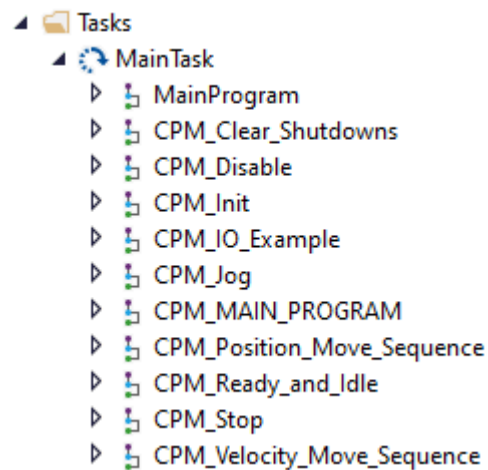
- 6. [optional] Import additional AOIs. The example .L5X will only import AOIs used by the Integration Ready example. To make all AOIs available in your program, import any missing AOIs.

DETAILED STATE MACHINE DESCRIPTION

The Integration-Ready example uses a state-based architecture, with each motor transitioning between states common in many machine applications (e.g.: Disable, Init, Ready/Idle, and various motion states).

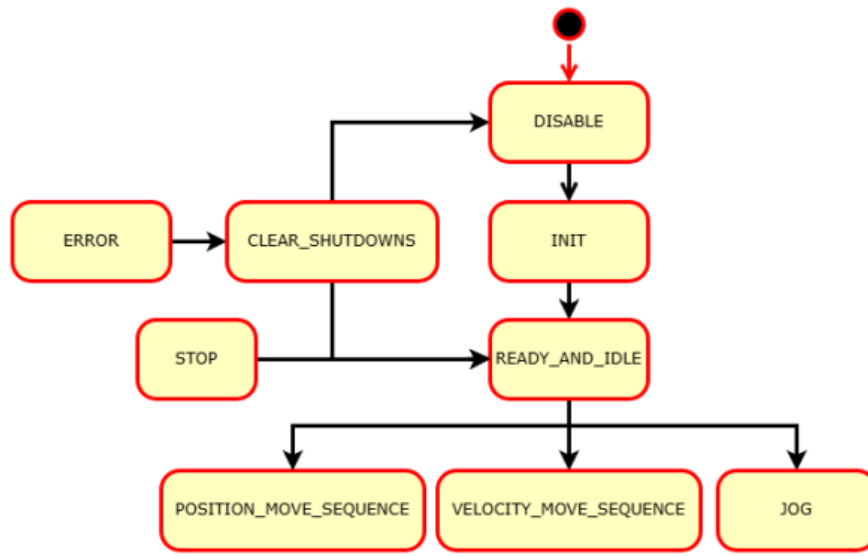
▸ motorState_CLEAR_SHUTDOWNS	70
▸ motorState_DISABLE	0
▸ motorState_ERROR	100
▸ motorState_INIT	10
▸ motorState_JOG	40
▸ motorState_POSITION_MOVE_SEQUENCE	30
▸ motorState_READY_AND_IDLE	20
▸ motorState_STOP	60
▸ motorState_VELOCITY_MOVE_SEQUENCE	50

Most states have an associated Program scheduled within the MainTask. The routines within these programs check the state of each individual motor and execute the logic associated with that state.



The user controls the desired motor states within the CPM_MAIN_PROGRAM. The rungs of this program typically put all motors into the same state (e.g. “RunVelocityMoveSequence” makes all motors jog back and forth), but you can customize these rungs so different motors are in different states; the program architecture allows for this.

Below is a simple state diagram showing the included default states:



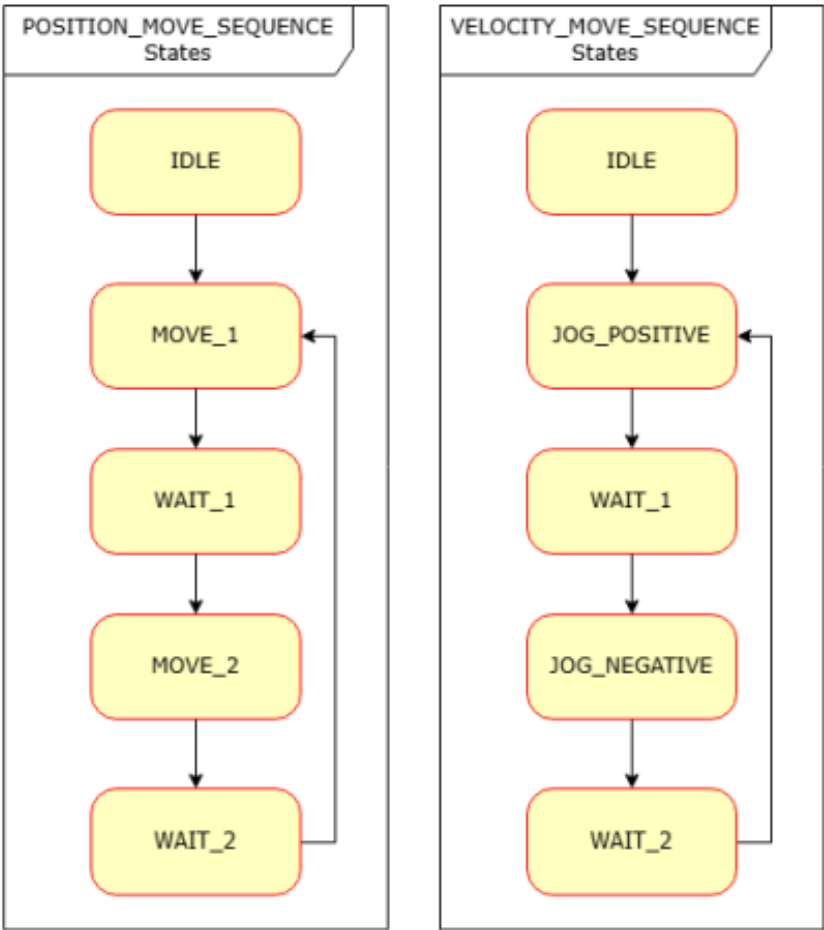
State	#	Description
DISABLE	0	This is the default state on power-up. While Disabled, the motor will not apply torque or hold any commanded position. Commanding this state during motion will immediately stop any executing motion.
INIT	10	This is the state that enables the motor before normal operation. If homing is required, it typically occurs during the INIT. The Motion Axis Home (MAH) AOI can be made to execute in this state.
READY_AND_IDLE	20	The motor is enabled and ready for any motion commands.
POSITION_MOVE_SEQUENCE	30	Moves the motor back and forth between two positions. This sequence will repeat until the STOP state is commanded.
JOG	40	Moves the motor continuously in the positive direction. Jogging continues until the STOP state is commanded. Optionally use hardware inputs to control direction.
VELOCITY_MOVE_SEQUENCE	50	Moves the motor back and forth using jogging velocity commands. This sequence will repeat until the STOP state is commanded.
STOP	60	Stop any executing motion and return to the READY_AND_IDLE state.
CLEAR_SHUTDOWNS	70	Attempt to clear motor shutdowns. Automatically transition to DISABLE or READY_AND_IDLE states if shutdowns were cleared successfully.
ERROR	100	This state is automatically transitioned to if any of the other states encounter an error or unexpected motor shutdown.

MOVE SEQUENCE DESCRIPTION

Each move sequence uses its own state machine of local state tags. See below for a state diagram of each sequence.

▸ MoveSequenceState_IDLE	Local	0
▸ MoveSequenceState_MOVE_1	Local	10
▸ MoveSequenceState_MOVE_2	Local	30
▸ MoveSequenceState_WAIT_1	Local	20
▸ MoveSequenceState_WAIT_2	Local	40

▸ MoveSequenceState_IDLE	Local	0
▸ MoveSequenceState_JOG_NEGATIVE	Local	30
▸ MoveSequenceState_JOG_POSITIVE	Local	10
▸ MoveSequenceState_WAIT_1	Local	20
▸ MoveSequenceState_WAIT_2	Local	40



HOW TO USE THE EXAMPLE

Toggle the Run tags (RunInit, RunJog, RunStop etc.) in the CPM_MAIN_PROGRAM routine to exercise the motors.

Use the CPM_IO_Example program to exercise the general-purpose I/O of the IO-HUB-4-E.

HOW TO MODIFY THE EXAMPLE

The Integration-Ready Example was designed to allow for customization to fit your application requirements.

For example, by changing only CPM_MAIN_PROGRAM, you can:

- Automate what triggers each motor state.
- Put each motor into different states at different times (e.g.: Jog only Motor 0 while Motor 1 is Stopped)

Within CPM_Position_Move_Sequence and CPM_Velocity_Move_Sequence, you can:

- Customize the positions and speeds of each motor
- Change the move sequence to command additional speeds or positions

With further edits, you can:

- Add custom motor states or additional move sequences.
- Control additional axes by adding more hubs and motors. Each hub requires its own Data Exchange AOI in the main program.

Click on the “Add-On” tab of the Language Element toolbar to easily access the ClearPath-IP AOIs while programing your PLC application.

