

Example Code Manual

Feed-to-length Application using **MC TouchProbe for Position Latching**

Applicable Product: MP2300Siec w/ MotionWorks IEC

Subject: Example Code Manual	Product: MP2300Siec and MotionWorks IEC	Doc#: EM.MCD.08.011
Title: Feed-to-Length Application using MC_TouchProbe for Position Latching		

Table of Contents

1. Application Overview:	3
a. Application Highlights	3
b. Application Challenges	3
c. Products Used:	3
2. Quick Startup Procedure (Project transfer to controller)	4
3. Program Explanation	5
4. Program Control Handles	7
5. Configuration of Latch (EXT 1)	8
6. Sequence of Operations	9
7. Results.....	10

Subject: Example Code Manual	Product: MP2300Siec and MotionWorks IEC	Doc#: EM.MCD.08.011
Title: Feed-to-Length Application using MC_TouchProbe for Position Latching		

1. Application Overview:

This document is intended to guide a user through the steps of demonstrating a feed-to-length application with the MP2300Siec controller in the MotionWorks IEC environment. The example code was written with MotionWorks IEC firmware version 0.9.2.2 and MotionWorks IEC Configuration Tool version 0.9.3.3. The demo tested had two axis (SIII master and SII slave). The feed axis was a Sigma II servopack axis.

a. Application Highlights

Industry:	Yaskawa customers wanting PLCopen Function Blocks for their applications
Major Features:	PLCopen compliant Function Blocks, IEC61131-3 standard programming environment
Results:	Feed-to-length application

b. Application Challenges

- Registration response consistency
-

c. Products Used:

Component	Product and Model Number
Controller	Yaskawa MP2300Siec controller with LIO-01 card
Hardware	Sigma II amplifier and motor
Software	MotionWorks IEC

Subject: Example Code Manual	Product: MP2300Siec and MotionWorks IEC	Doc#: EM.MCD.08.011
Title: Feed-to-Length Application using MC_TouchProbe for Position Latching		

2. Quick Startup Procedure (Project transfer to controller)

- I. Go to the 'Hardware' tab. (Fig 1)
- II. Right click 'Resource: MP2300Siec' (Fig 1)
- III. Under 'Settings', set the IP address of the controller
- IV. Make the project (Fig 2)
- V. Download the project to the controller through the resource icon (Fig 3)
- VI. Set in Debug mode(Fig 3)

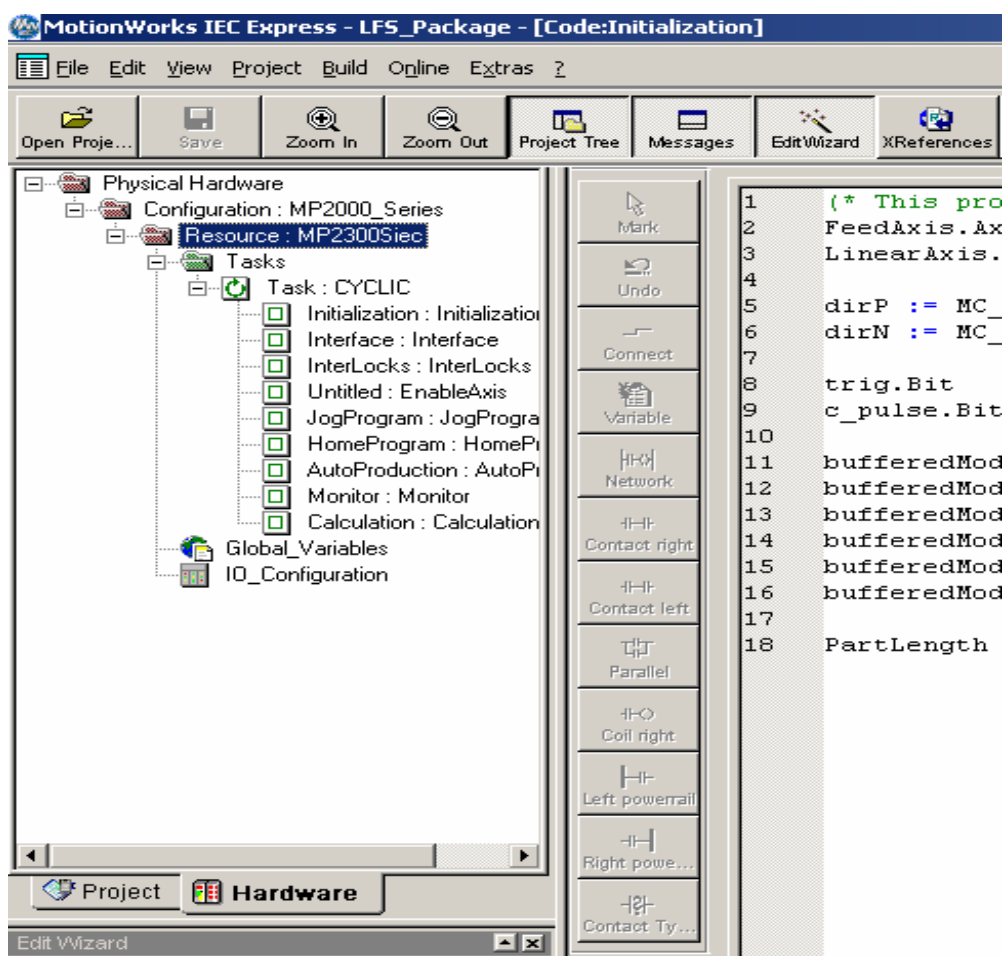


Figure 1: Setting IP address

Subject: Example Code Manual	Product: MP2300Siec and MotionWorks IEC	Doc#: EM.MCD.08.011
Title: Feed-to-Length Application using MC_TouchProbe for Position Latching		

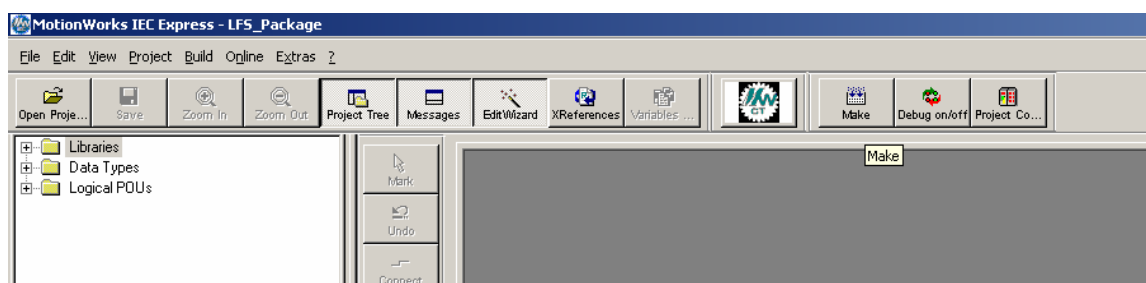


Figure 2: Make

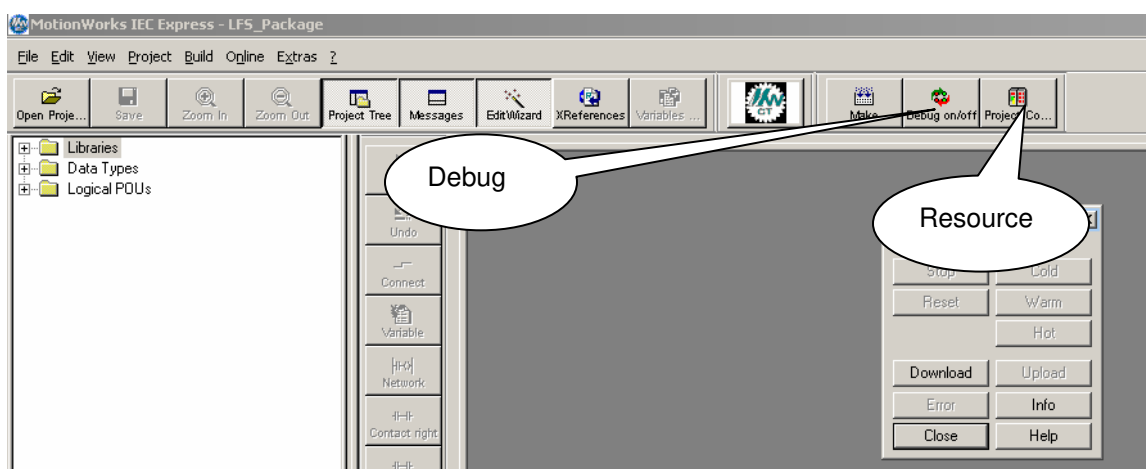


Figure 3: Download and debug

3. Program Explanation

The project consists of 7 programs. They are:

1. **Initialization:** This program initializes the axis, the trigger variable for latching, the length of the part being used in the feed routine, the variable definition for direction, and the buffered modes.
Attribute: Makes use of structured text for definitions

Subject: Example Code Manual	Product: MP2300Siec and MotionWorks IEC	Doc#: EM.MCD.08.011
Title: Feed-to-Length Application using MC_TouchProbe for Position Latching		

2. **Interface:** This program assigns the 6 inputs to the input variable names. The two outputs are also defined in this program.
Attribute: Makes use of structured text for definitions
3. **Interlocks:** This program enables the various modes of operation. The two main modes are manual and automatic. In manual mode, the axis can be jogged and homed to the C channel of the encoder. In automatic mode, the axis can be used for part production. The interlocks for various modes, and how they synchronize with the inputs and outputs, can be seen in this program.
Attribute: Makes use of structured text for definitions
4. **EnableAxis:** This program controls the axis enabling sequence. The axis gets enabled only if there are no axis errors. This program also controls clearing any alarms that may be present on the axis before enable is carried out.
Attribute: MC_Power, MC_ReadAxisError, MC_Reset
5. **JogProgram:** This program provides function blocks for the jog routine.
Attribute: MC_MoveVelocity, MC_Stop
6. **HomeProgram:** This program controls the home routine, which can be activated in manual mode. The axis is driven in a negative direction, until a marker on the axis belt is seen. The axis is stopped and after a time delay, the axis moves to the closest c channel and stops.
Attribute: MC_MoveVelocity, MC_Stop, MC_StepRefPulse
7. **AutoProduction:** This program controls the automatic production. On enabling the auto mode and starting production, the axis continues motion until a part is sensed. The axis location is then latched. The first motion is aborted and a new relative motion is executed, bringing the axis to rest at a desired distance after the axis latch position.
Attribute: Uses ladder logic for various interlocks. Uses rising and falling edge detectors, timers. MC_MoveRelative, MC_TouchProbe

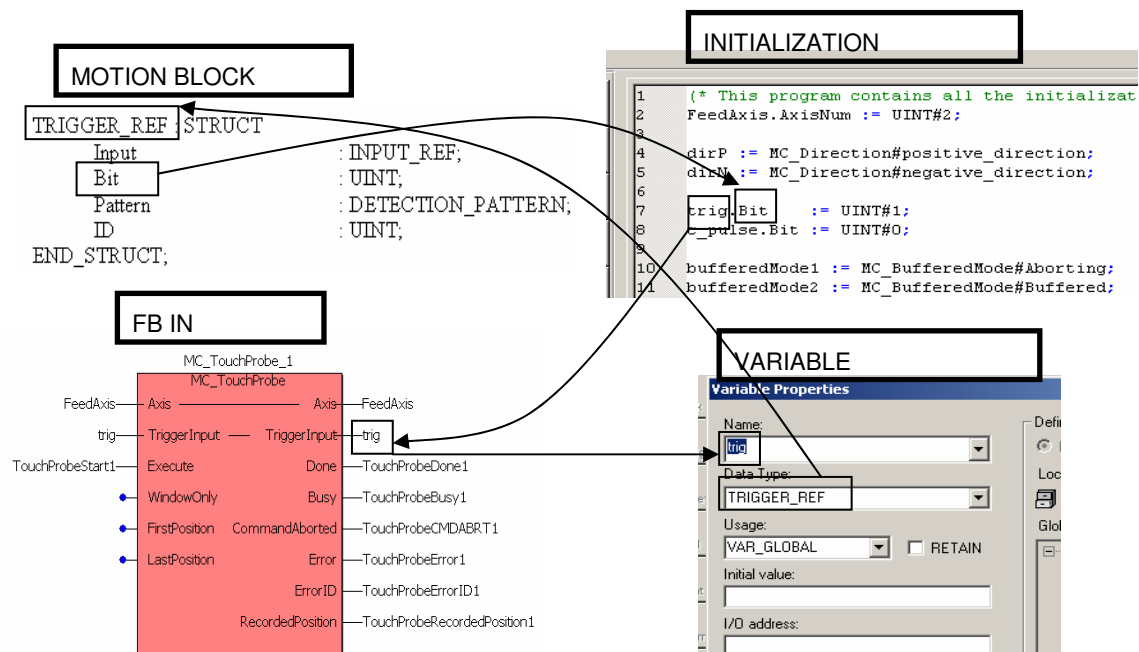
Subject: Example Code Manual	Product: MP2300Siec and MotionWorks IEC	Doc#: EM.MCD.08.011
Title: Feed-to-Length Application using MC_TouchProbe for Position Latching		

4. Program Control Handles

- 1) Output 2 in a high state indicates an alarm on the axis. This can be cleared using Input 6, which resets the alarm. If reset fails, controller/servopack power will have to be cycled.
- 2) If there are no alarms, Input 1 enables the feed axis.
- 3) Input 2 in a low state indicates manual mode and jogging or homing can be accomplished.
- 4) Input 3 jogs the axis and does so until input 3 is turned off.
- 5) Input 4 in a high state starts the homing routine. The axis runs until it encounters a signal on the EXT 1 input of the servopack. At this signal, the axis stops. After a delay, the axis moves slowly to stop at the C channel position of the axis encoder. After homing is completed, turn input 4 off.
- 6) Input 5 starts automatic production. However, for auto production to start, the controller should be in auto mode (Input 2 high). Also, auto start is possible only after the homing routine has been accomplished. Autostart (Input 5) gets the axis moving. On encountering a part (latched EXT 1 signal), the move is aborted by another relative move. The axis then moves a predefined distance and stops. The axis restarts the cycle after a small predefined pause. This continues until input 5 is turned off. If no part is encountered after a predetermined variable 'part length', Output 1 turns on indicating an alarm. This can be cleared, so the cycle can be restarted by cycling Input 5.

5. Configuration of Latch (EXT 1)

- 1) Set Pns 820, 821 (Latching area upper limit: 820:FFFF, 821:7FFF)
- 2) Set Pns 822, 823 (Latching area lower limit: 822:0000, 823:8000)
- 3) Set Pn511 to match wiring (CN1)



Subject: Example Code Manual	Product: MP2300Siec and MotionWorks IEC	Doc#: EM.MCD.08.011
Title: Feed-to-Length Application using MC_TouchProbe for Position Latching		

6. Sequence of Operations

Cold start the application and activate debug mode:

- 1) If there are axis alarms, clear them using **Switch 6**. If the alarms cannot be cleared using Switch 6, power will have to be cycled.
- 2) Use **Switch 1** for enabling both axes.
- 3) Make sure **Switch 2** is low, ensuring that the machine is in manual mode at first.
- 4) **Switch 3** jogs the feed axis
- 5) **Switch 4** starts the homing sequence. The feed axis homes to a latch sensor first, and then comes back to the closest C channel pulse. To simulate the latch, use the LATCH switch on the demo corresponding to the SGDH axis. After a small time delay after the latch-induced stop, the feed axis moves to C pulse and sets position as zero (home). Turn switch 4 off after the axis completes homing.
- 6) To start auto production, the axis has to be homed. Turn **Switch 2** and **Switch 5** on. To start at its production velocity, use the LATCH switch, as this enables the feed-to-length algorithm. The axis stops after a pre-decided distance beyond the feed axis latched position. It then waits for a press/cut operation, and starts moving again looking for the next part latch. Auto cycle can be stopped by switching Switch 5 off.

7. Results

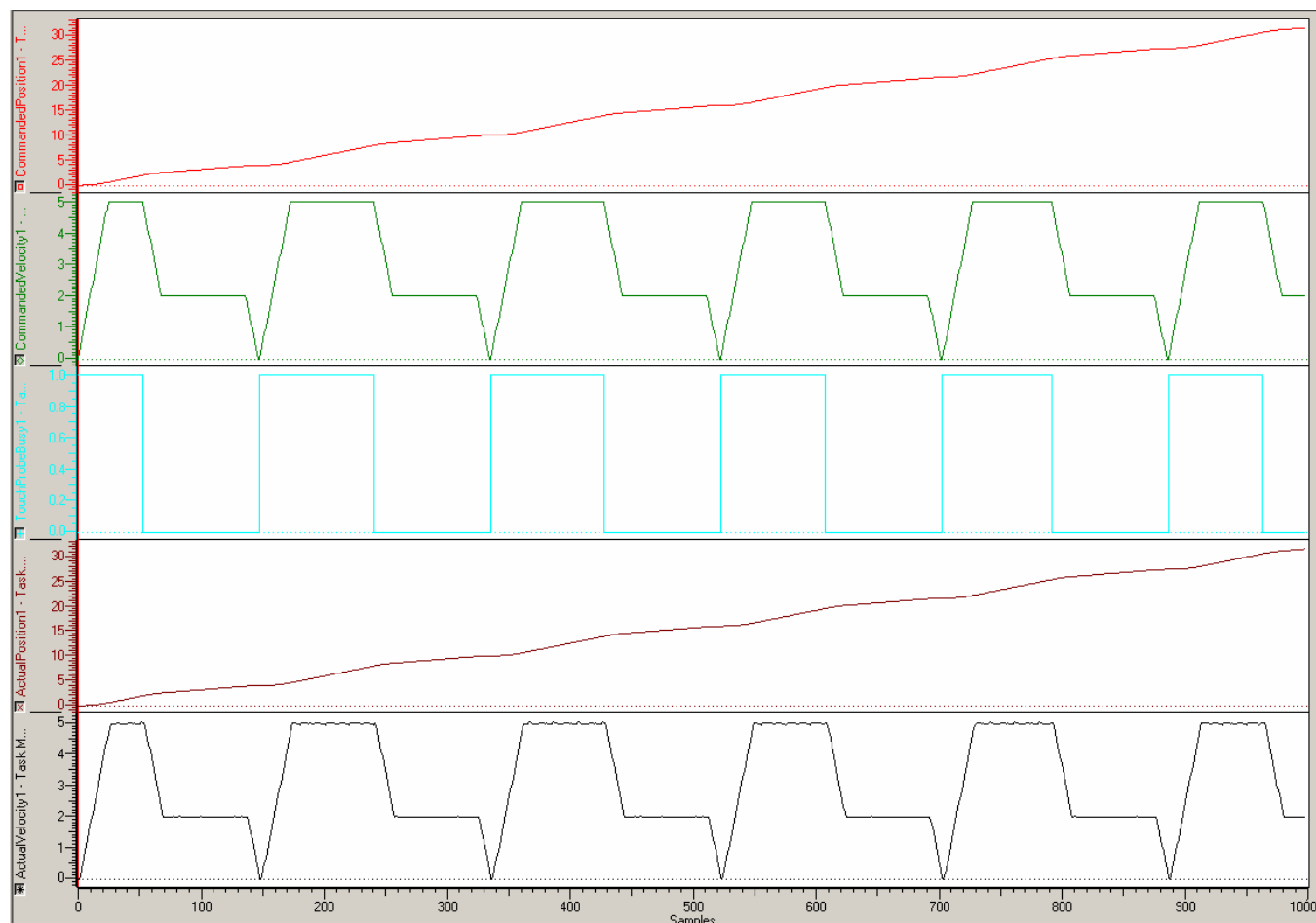


Figure 4: [traces from top to bottom]

- a) Commanded position
- b) Commanded velocity
- c) Touch Probe busy
- d) Actual Position
- e) Actual Velocity