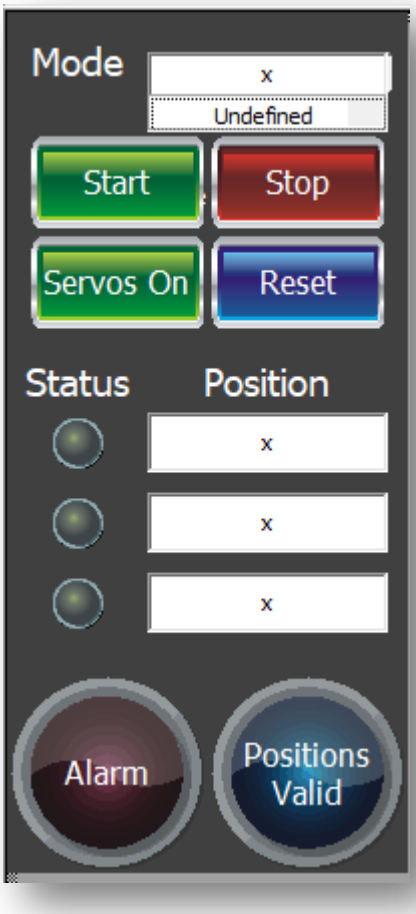
YASKAWA

Control Panel

Mini-Lab

Requirements

- HMI Project created with placeholders for all screens



Lab Overview

This lab document will guide the participant through the following steps:

I. Arrange and Format Controls (10 min)
II. Set Object Properties (35 min)
III. Verify Functionality/Troubleshoot (15 min)

Lab Goal

- The control panel is fully operational

I. Arrange and format the controls on the screen (15 min)

Copy from existing screens to save time and give consistent look and feel

A. Recommend button height: 50 (for finger operation)

B. Text and background color as desired

C. For Mode use a ComboBox (Toolbox → Objects → ComboBox)

II. Set Object Properties

A. Refer to the following table of variables

Variable	Description
HMI_ActiveScreen	_SysVar_:ActiveScreen from HMI
HMI_ControllerAlarm	Status of MPiec controller Alarm output (ALM led)
HMI_ControlMode	MPiec "mode" of operation. 0=invalid, 1=production, 2=maintenance, 3=manual
HMI_StartRequest	Start the production mode sequence
HMI_SvOnRequest	Request all axes Servo On
HMI_X_DriveAlarm	Status of SERVOPACK Alm
HMI_X_CurrPos	Actual Position
HMI_X_ServoOn	Servo is On (SVON)
HMI_X_StopRequest	Stop and prohibit all motion
HMI_Y_DriveAlarm	Status of SERVOPACK Alm
HMI_Y_CurrPos	Actual Position
HMI_Y_ServoOn	Servo is On (SVON)
HMI_Y_StopRequest	Stop and prohibit all motion
HMI_Z_DriveAlarm	Status of SERVOPACK Alm
HMI_Z_CurrPos	Actual Position
HMI_Z_ServoOn	Servo is On (SVON)
HMI_Z_StopRequest	Stop and prohibit all motion

B. Servos On

Toggles all servos on and off

1. Command/state variable **HMI_SvOnRequest**
2. Command Type = ON-OFF

C. Reset

Resets production mode part counter

1. Command Type = Impulsive
2. Impulsive time = 800 (means the variable stays true for exactly that amount of time)
3. Command/State Variable = **HMI_Reset**

D. Status

Displays the Servo On status for each axis

1. Command/state Variable = **HMI_<axis>_ServoOn** (per axis)

E. Position

Displays the position of each axis in units of [mm]

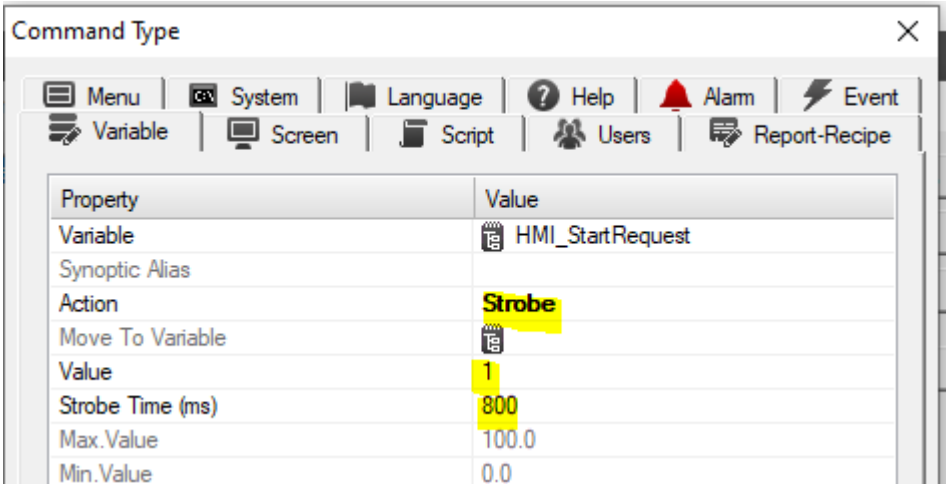
1. Style➔ Read Only = True
2. Style➔Format Value = x.xx
3. Style➔Format Engineering unit = mm
4. Min.Value and Max.Value have no effect when Read Only is selected
5. Editbox-Display Variable = **HMI_<axis>_CurrPos**

F. Start

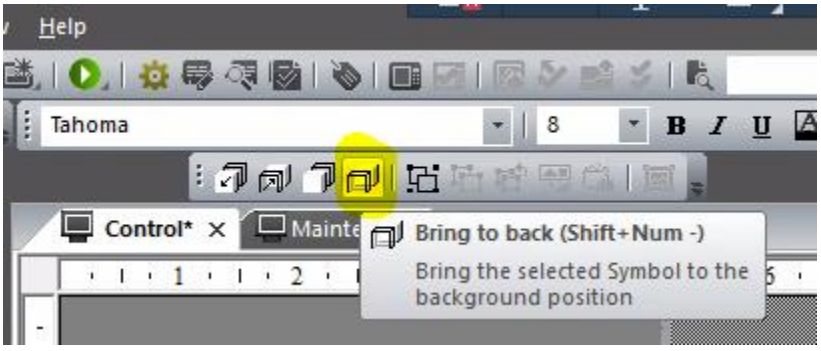
Starts the 3-axis move sequence when MPiec is in production mode

Button should be visible only when ControlMode = 1 (production mode)

1. Properties → Execution →
   i. Command/State Variable = **HMI_Start_CMD**
   ii. Command Type = Execute Commands
   iii. Commands on Release = Strobe **HMI_StartRequest** to value=1 for 800ms



2. Properties → Dynamics → Visible →
   i. Enable Visibility = true
   ii. Variable = **HMI_ControlMode**
   iii. value=1 (for mode 1)
3. Enter text box “Start only in production mode”
   i. “Bring to back” behind the start button

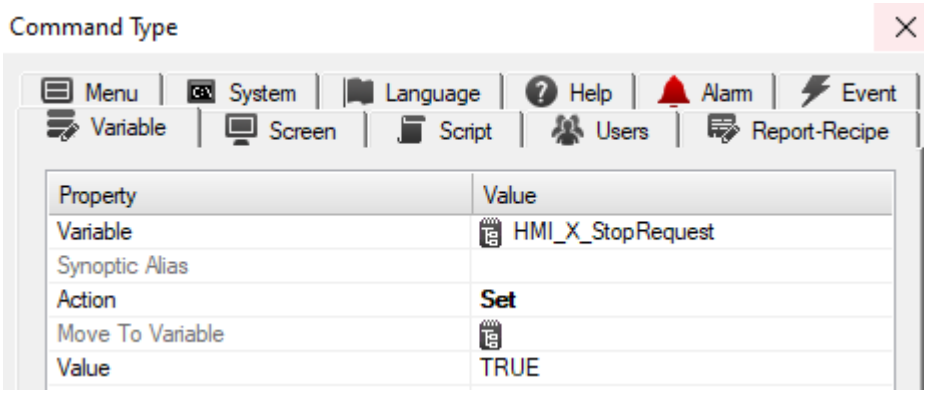


G. Stop:

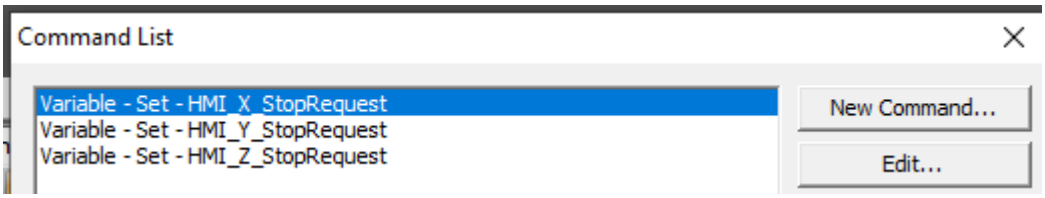
Stops all motion on all servo axes

Each servo axis uses individual variables for stop command

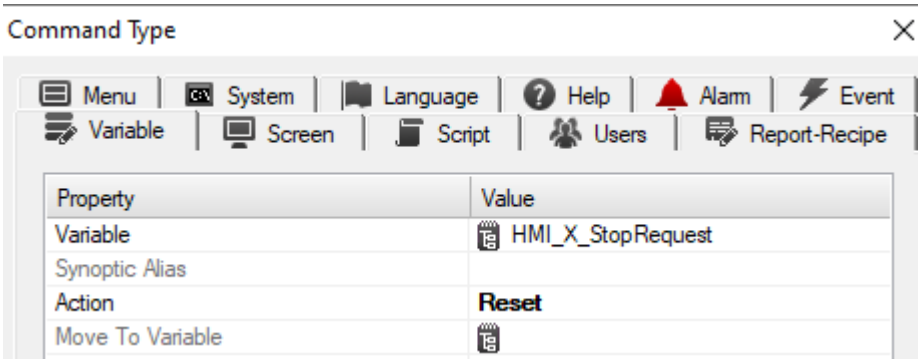
1. Cannot use Command/State Variable to set multiple variables in one button
2. Properties → Execution →
    i. Command Type = Execute Commands
    ii. Commands on Pressed = set HMI_<axis>_StopRequest = 1 (true) for each axis



3. Use copy, paste, edit for each axis



4. Commands on Release = reset stop request for each axis
    i. Properties → Execution



5. Properties → Execution → Advanced → Impulsive Time = 800 [ms]

    Impulsive time <> 0 allows a button with “Execute Commands” to behave like a push button.

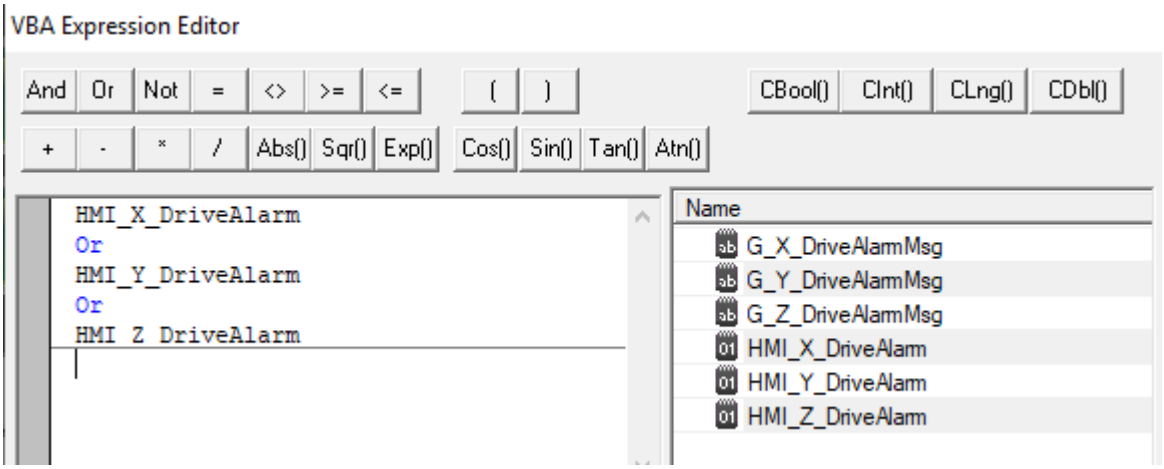
    Impulsive time sets the exact time of the ON pulse.

H. Alarm

Control illuminates if any of the axes report an alarm

The alarm variables are HMI_<axis>_DriveAlarm

1. Command/State Variable = Expression
2. Use “OR” logic for all alarm variables from each axis
    i. **HMI_<axis>_DriveAlarm**
3. set filter `*alarm*` then build expression

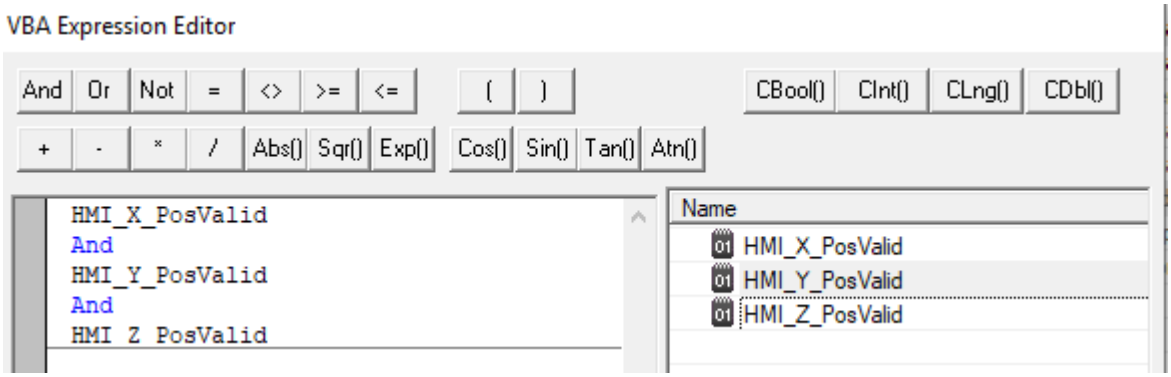


I. Positions Valid

Indicates that all axes have been homed or zero point was set

The position is validated and absolute moves are possible

1. Command/State Variable = Expression
2. Use “AND” logic for all alarm variables from each axis
    i. **HMI_<axis>_PosValid**
3. set filter `*posvalid*` then build expression
4. Use AND logic for all axes for all PosValid



J. Mode (combo box)

- ***Displays the control mode of the MPiec***
- ***MPiec sends a numeric value for Control Mode to the HMI: 0=Undefined, 1=Production, 2=Maintenance, 3=Manual***
- ***MPiec determines control mode based on HMI_ActiveScreen along with other logical conditions.***

1. Properties → Variables
    i. EditBox-Display Variable = **HMI_ControlMode**
2. Properties → Style
    i. Listbox Items = Undefined | Production | Maintenance | Manual
        a. Type the items. Do not use "…" as this brings up the string table (for mult-language support)
        b. Items separated by pipe symbol 0|1|2|3|…|n according to numeric order starting at zero (like an Enumerated Type in MotionWorks IEC)
    ii. Spin Size = Custom (to allow spin width = 0)
    iii. Spin Width = 0 (to remove drop-down control)
    iv. Max Value = 3 (mode 3=Manual is the highest value)

III. Create the Event to control **HMI_ActiveScreen**

Notes

- ***The active screen variable in the MPiec (HMI_ActiveScreen) must always be the same as the active screen variable in the HMI (_SysVar_:ActiveScreen)***
- ***If they are not the same, then use an event to write the value from HMI to MPiec***

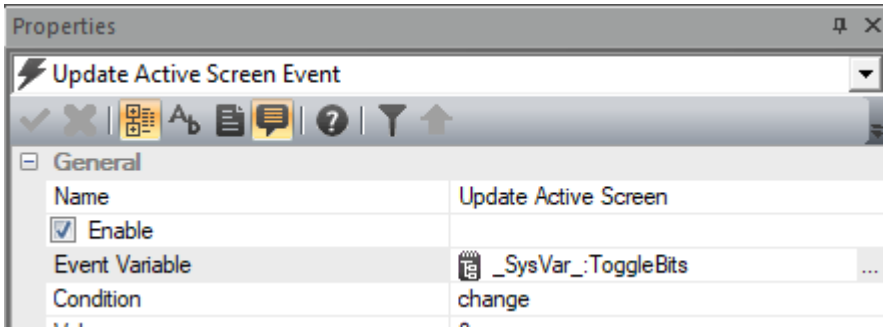
A. Create a new event named `Update_ActiveScreen`

1. Right-click to "Add New Event Object"

B. Set Event Variable to _SysVar_:ToggleBits

Event will move the value of* _SysVar_:ActiveScreen *to* HMI_ActiveScreen *when_SysVar_:ToggleBits changes. (These bits constantly change!)

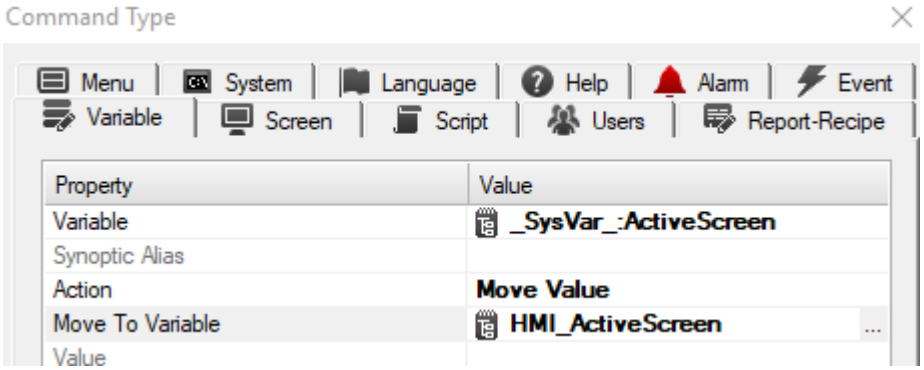
1. Properties → Event Variable → "…" →
2. Find the SysVar



C. Event Commands → "…" → New Command

***The "Toggle Bits" are constantly changing. Use this to constantly move the value of* _SysVar_:ActiveScreen *to* HMI_ActiveScreen**

1. Variable = _SysVar_:ActiveScreen
2. Action = Move Value
3. Move To Variable = HMI_ActiveScreen



IV. Verify Operation

When using the runtime, first stop the HMI hardware so it doesn't interfere

Verify operation in runtime using "watch" and "Locals"

Verify operation on the smartPanel using the variables in the Web UI

A. Mode indicator reflects the correct mode as you navigate between screens

1. Troubleshoot functionality of the event that controls **HMI_ActiveScreen** using the MP3300iec Web UI

B. Servos turn on and off (drive LED display changes)

C. Status indicators reflect status of servo on

D. Start button appears only on Auto and Recipe screen

E. To start motion

1. Auto screen
2. Mode = Production
3. Reset (cycle counter)
4. Servos On
5. Start – button stays on. Position feedback indicates each motor moves 90mm and return to zero.

F. Alarm indicator functions

1. Turn on the "Low PE" button in the Setup screen
2. Attempt to start motion
3. Alarm indicator lights
4. Servos display A.d00
5. Clear Alarms in Setup screen

End Of Mini-Lab

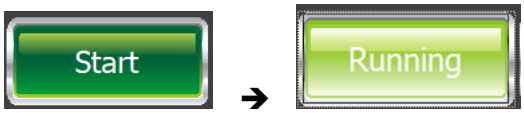
Troubleshooting Tips

- Mode: Be sure to enter text in Style→ListBox Items and not Variables→ListBox List Variable
- Mode: Use watch and WebUI to monitor _SysVar_:ActiveScreen and HMI_ActiveScreen and HMI_ControlMode

Certification Checklist

- Mode changes depending on active screen
- Servos turn on/ status on
- Position feedback for each axis with 2 digits after decimal
- Reset cycle counter to zero
- Start button appears only in Production mode
- Start button illuminates only if motion started, not if zero cycles remaining.
- Alarm indicator turns on with alarm
- Pos Valid indicator turns on when all axes valid

V. (Optional) Make the Start button read “Running” when illuminated

A. 

B. Properties-->Dynamics--> Dynamic Text - Edge Color.

1. Select "enable…"
2. Copy /paste the variable **HMI_Start_CMD** from command state
3. Open Edit Text-Edge Colors.
    i. Set the color and Dynamic Text for the variable value threshold.