

## Auto Screen

### Mini-Lab

#### Requirements

- MP3300iec Demo
- Functional screens for Jog-Zero, Setup, control panel, navigation bar
- JPG image file representing the product

#### Lab Overview

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

- Build the Screen Layout (15 min)
- Assign Object Properties (30 min)
- Verify Operation (15 min)

#### Lab Goal

- Data can be entered and displayed on the Auto screen
- Motors move according to data in Auto screen
- (Optional) Image of product moves along conveyor symbol in real-time according to motor position



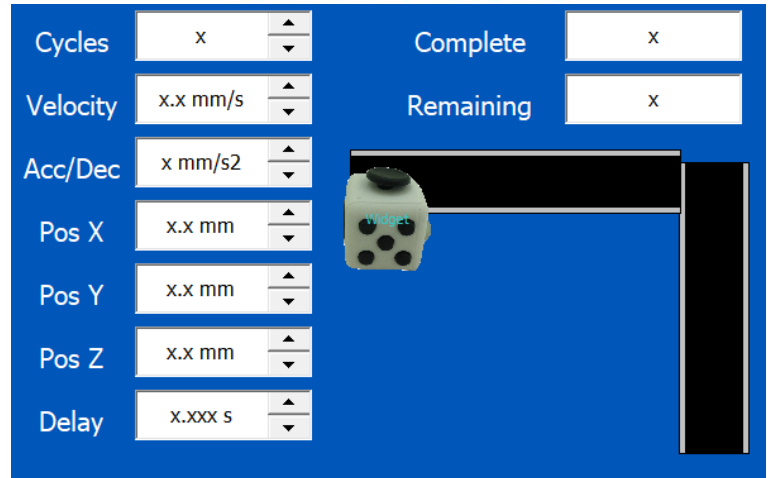
## I. Build screen layout (15 min)

### A. Enter controls

1. Text
2. Editbox-Display
3. Remember the Layout menu

### B. Enter symbols

1. Conveyor
2. Provided "widget" image file or other image or symbol
3. View → Symbol Libraries



## II. Assign Object Properties (30 min)

### A. Cycles, Complete, Pos X, Pos Y, Pos Z, Delay

1. Refer to the table of MPiec variables

### B. Velocity & AccDec EditBox-Display

1. Use new HMI variables **MoveVel** and **MoveAccDec**
  - i. Type = Float (32-bit)
  - ii. Set Initial Value
  - iii. Retentive not Shared (so it retains value at power cycle)
2. Create a new event to move the value of **MoveVel** to each axis variable
  - i. New event named `Update_MoveVel`
  - ii. If speed change detected, set commands to "Move Value" into the MPiec **MoveVel** variables for each axis
  - iii. Example of the command

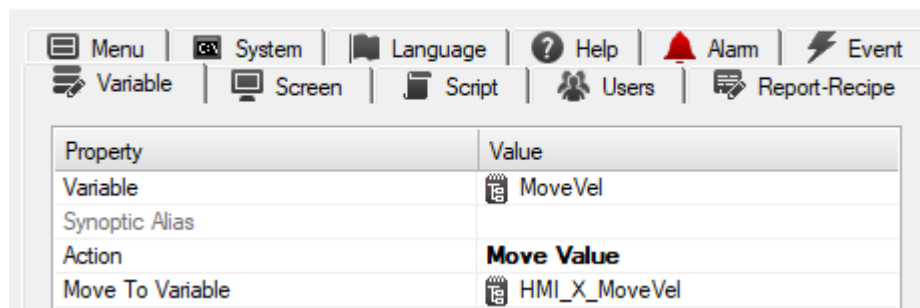
#### Variable

HMI\_CycleTarget  
HMI\_Delay  
HMI\_CycleComplete  
HMI\_X\_MoveABSPos  
HMI\_X\_MoveAccel  
HMI\_X\_MoveDecel  
HMI\_X\_MoveVel  
HMI\_Y\_MoveABSPos  
HMI\_Y\_MoveAccel  
HMI\_Y\_MoveDecel  
HMI\_Y\_MoveVel  
HMI\_Z\_MoveABSPos  
HMI\_Z\_MoveAccel  
HMI\_Z\_MoveDecel  
HMI\_Z\_MoveVel

#### Description

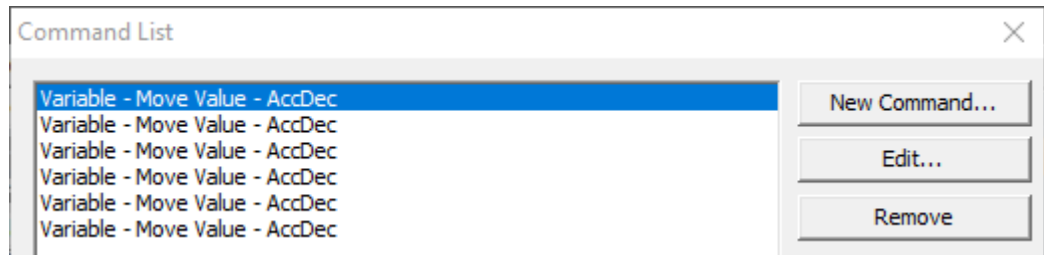
Target number of cycles  
number of ms between moves  
Number of cycles that have completed  
Absolute target position  
Acceleration rate for positioning  
Deceleration rate for positioning  
Velocity (Speed) for positioning  
Absolute target position  
Acceleration rate for positioning  
Deceleration rate for positioning  
Velocity (Speed) for positioning  
Absolute target position  
Acceleration rate for positioning  
Deceleration rate for positioning  
Velocity (Speed) for positioning

#### Command Type



3. Repeat to create event `Update_AccDec` which writes **AccDec** to **MovAcc** and **MovDec** variables for each axis

- i. Command type similar to `MoveVel`

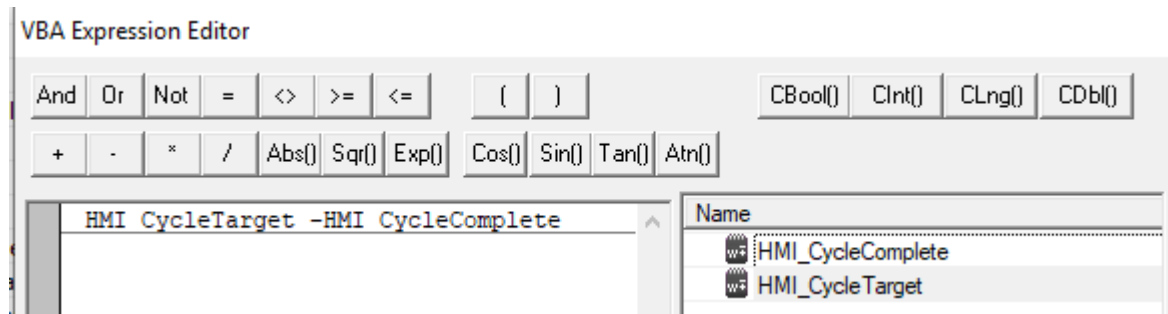


## C. Delay

1. Time datatype in the MPiec appears as `DWORD` in Movicon
2. Variables with `DWORD` datatype can be displayed with a decimal point. All properties, however, are still entered as integers (Min.Value, Max.Value, Spin Step, etc)
  - i. Format Value = x → User enters 12345 → 12.345 [s] in MPiec
  - ii. Format Value = x.xx → User enters 123.45 → 12.345 [s] in MPiec
  - iii. Conclusion – use format value x [ms] or x.xxx [s]
3. Resolution is 4ms (fast task cycle) so set Spin Step = 4 [ms]

## D. Remaining

1. `edTextBox-Display Variable` = Expression using `Cycle Target` and `Cycle Complete` from the MPiec to calculate remaining cycles.
2. A new variable is not required.



3. Style = Read Only

## III. Verify Operation (15 min)

A. To Operate: enter data in left column, reset, servo on, start

1. Set zero (Positions Valid)
2. Reset
3. Set data on the left column
4. Servos On
5. Start
6. Observe right column
7. Observe motors

B.

C. Add variables to watch to confirm value

D. Check for copy paste errors

E. Data field formatting

F. Change data in fields and expect motor response

G. Do Max/Min value make sense?

H. Is Spin Step a useful value?

## End Of Mini-Lab

### Troubleshooting Tips

- ☐ Check for copy-paste errors
- ☐ Confirm EditBox-Display formatting such as Max/Min value and Spin Step
- ☐ Add variables to watch to confirm the events are working

### Certification Checklist

- ☐ Cycles, complete, and remaining have suitable format and correct function
- ☐ Speed and Acc/Dec with suitable decimal point and max
- ☐ Pos X,Y,Z with reasonable range including negative values
- ☐ Delay displays correctly as ms or s
- ☐ Symbols represent the machine layout. Animation is optional.
- ☐ All fields affect application function (faster, slower, further)




## IV. (Optional) Visualization of moving widget Z movement

### A. Enter widget image (or any image, or an image from the Symbol Libraries)

1. Toolbox → Basic Shapes → Rectangle
2. Stroke Attributes → Border size = 0
3. Background Attributes → Brush Style = Null, Static Image, stretched, keep proportions
  - i. Prefer PNG, JPG

### B. X and Y movement


1. Part to appear on HMI only when MPiec detects part present
  - i. Dynamics → Visible
  - ii. Enable Visibility: variable = **HMI\_PartPresent**
2. Part moves on HMI as motors turn
  - i. Dynamics → Move Horizontal (X) and Dynamics → Move Vertical (Y)
  - ii. Enable X Movement and Enable Y Movement
  - iii. Variable for Move X: use expression →
    - a. example: **HMI\_X\_CurrPos/ HMI\_X\_MoveABSPos\*160** (to move 160 pixels by the end of the move)
    - b. “Error has been caught...” window appears. Click YES.
  - iv. Set End Value X and End Value Y to the number of pixels to move. This is the max value for the expression.
3. The starting position and the “invert direction” selection may be required. Adjust this after testing.

[-] Dynamics	
[-] Visible	
<input checked="" type="checkbox"/> Enable Visibility	
Variable	 HMI_PartPresent
Value	1
Tolerance	0
Condition	equal
Blink	None
[+] Composited Movement (XY)	
[-] Move Horizontal (X)	
<input checked="" type="checkbox"/> Enable X Movement	
<input checked="" type="checkbox"/> Invert X Direction	
Variable for Move X	 HMI_X_CurrPos /HMI_Y_MoveABSPos *160
Start Value X	0
End Value X	160
Offset for the X Start Value	0
Offset for the X End Value	1
[-] Move Vertical (Y)	
<input checked="" type="checkbox"/> Enable Y Movement	
<input checked="" type="checkbox"/> Invert Y Direction	
Variable for Move Y	 HMI_Y_CurrPos/HMI_Y_MoveABSPos *130
Start Value Y	0
End Value Y	130
Offset for the Y Start Value	0
Offset for the Y End Value	1

## C. Z-axis movement

*Represent positive Z axis movement as zoom from 100% to 0%*

1. Enable Dynamic Scaling
2. Scaling Variable: use expression  $\rightarrow 100 - (\text{HMI\_Z\_CurrPos} / \text{HMI\_Z\_MoveABSPos} * 100)$
3. Min. Percent Scale

Scaling	
<input checked="" type="checkbox"/> Enable Dynamic Scaling	
Scaling Variable	 $100 - (\text{HMI\_Z\_CurrPos} / \text{HMI\_Z\_MoveABSPos} * 100)$
Start Value for Scaling	0
End Value for Scaling	100
Min. Percent Scale	<b>50</b>
Max. Percent Scale	100