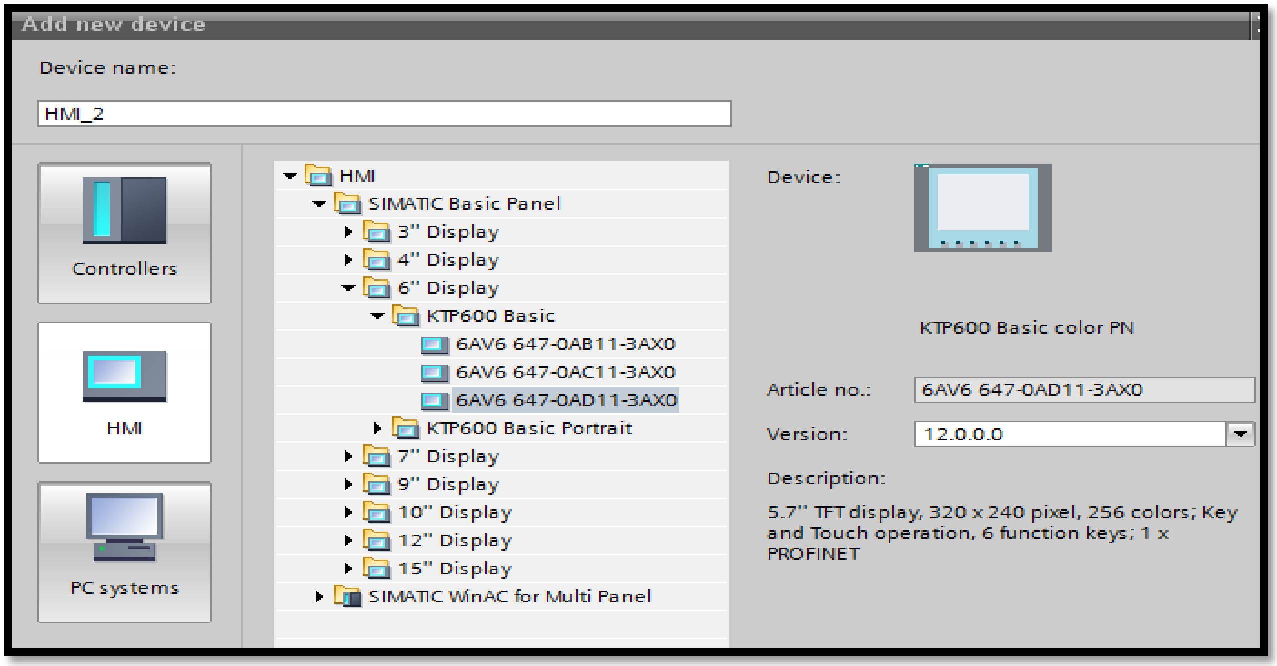
Lab 6: Operating Human Machine Interface Part 1

Now that we have learned the basics of coding and operating simple tasks, in this lab, we will learn how to add and code a Human Machine Interface screen and do basic design on the screen.

To Begin, we will need to add an HMI screen to the project.

Adding an HMI Screen:

1. Open your project file and enter the Device & networks tab
2. Select Add new device and choose the HMI option and name it HMI_1
3. In the HMI drop down menu, select SIMATIC Basic Panel
4. From there, select 6" Display, then KTP600 Basic
5. Select "6AV6 647-0AD11-3AX0", version 12.0.0.0



Device name: HMI_2

Article no.: 6AV6 647-0AD11-3AX0

Version: 12.0.0.0

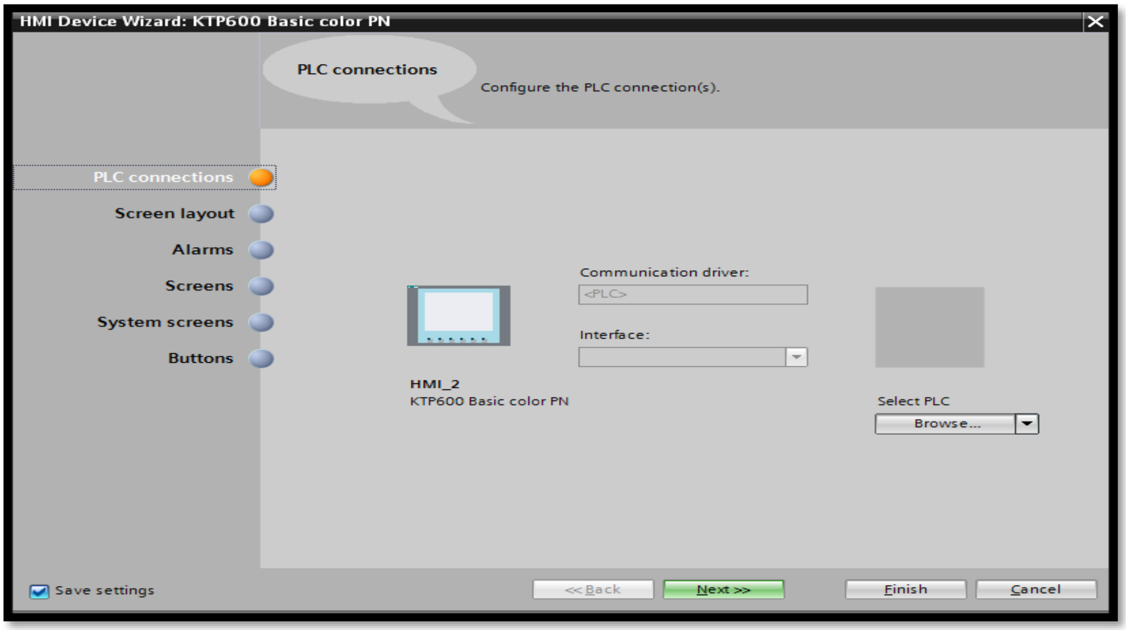
Description:

5.7'' TFT display, 320 x 240 pixel, 256 colors; Key and Touch operation, 6 function keys; 1 x PROFINET

Click Next and the HMI wizard will appear. There are 6 required steps must be completed in this wizard set up (see below).

Communication: This option allows you to create a connection with a PLC in which you will be operating with.

Click Browse to find the PLC that HMI will connect to. On the drop down choose PLC1 and connect. This is the most important step, so choose the right PLC for your project. Click next.



HMI Device Wizard: KTP600 Basic color PN

PLC connections

Configure the PLC connection(s).

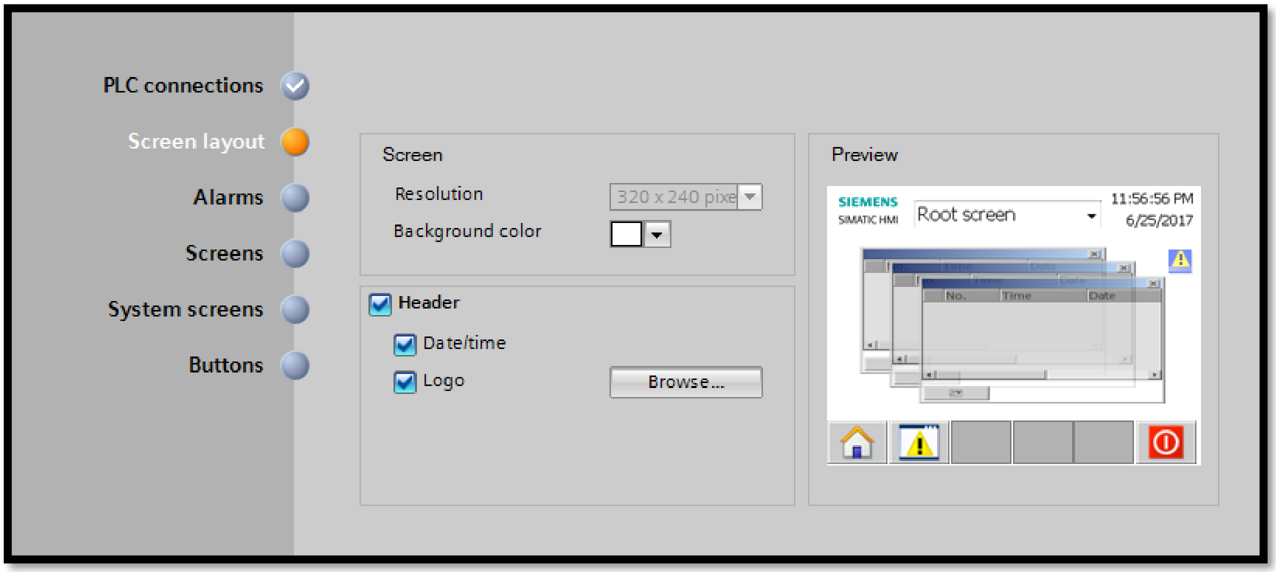
Communication driver: <PLC>

Interface:

HMI_2
KTP600 Basic color PN

Screen Layout: Here you can select the background color and headers.

White tends to work better as a background color. So select White and check both of the Headers. Once done, click Next.



Screen

Resolution: 320 x 240 pixe

Background color: □

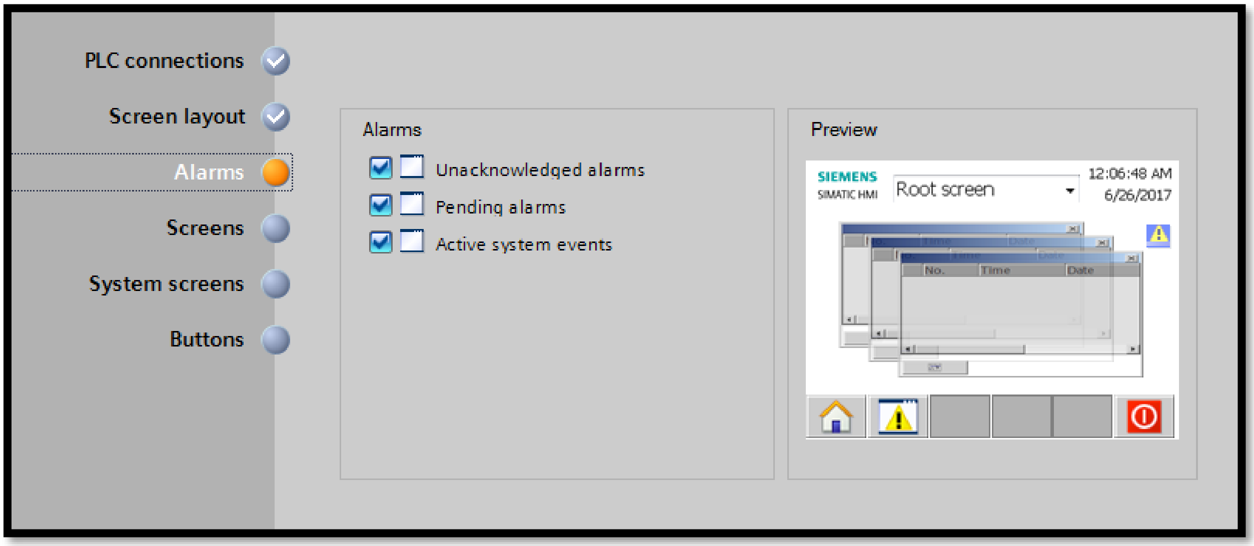
☑ Header

☑ Date/time

☑ Logo   Browse...

Alarms: Here you can select type of alarms you would want to see.

Select all 3 options then click next.



Alarms

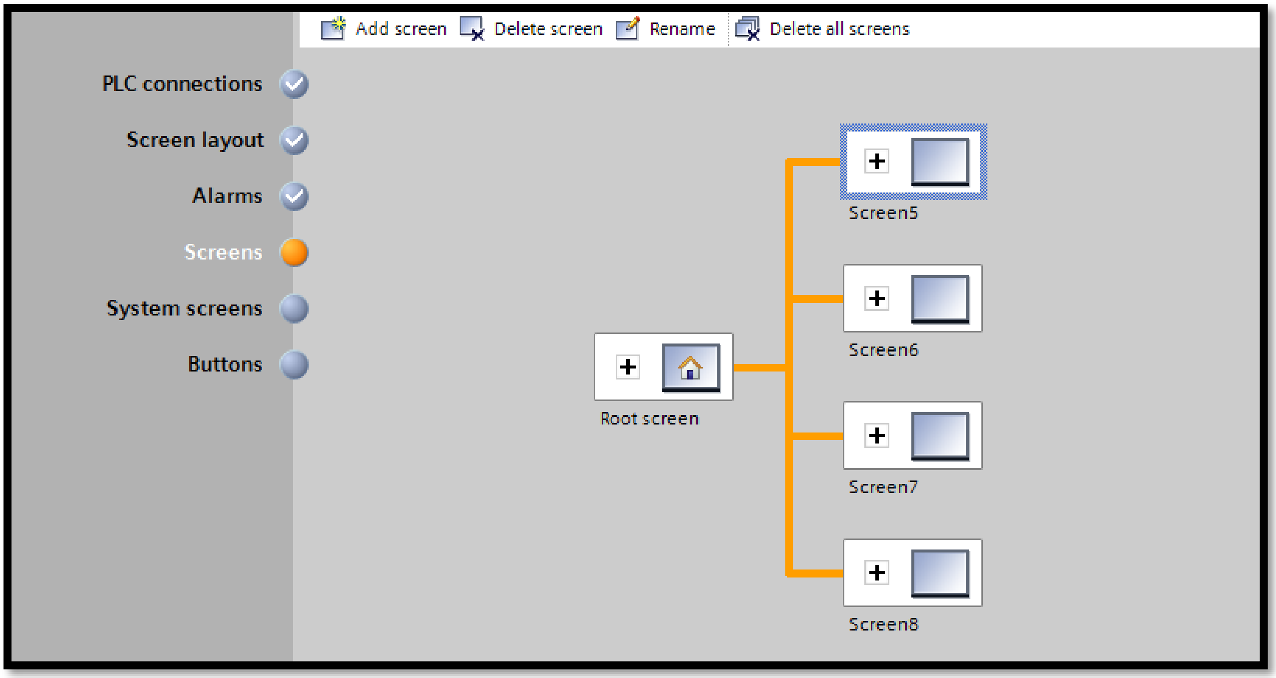
☑ Unacknowledged alarms

☑ Pending alarms

☑ Active system events

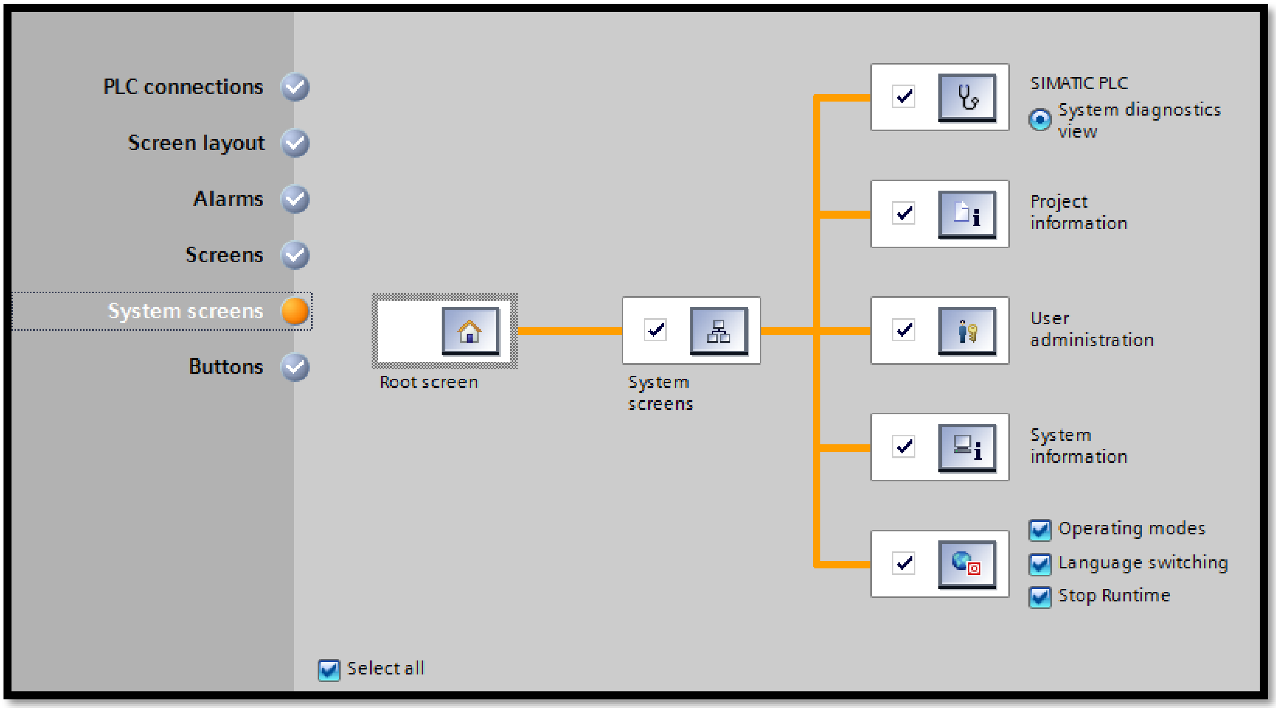
Screens: Here you can create add more screens and sub-screens

We want 9 sub screens from the root screen. Click the plus [+] button located at root screen 9 times.



System screens: System screens are pre made screens that has information about your project.

Select all screens to be displayed.



● System diagnostics view

☑ Operating modes

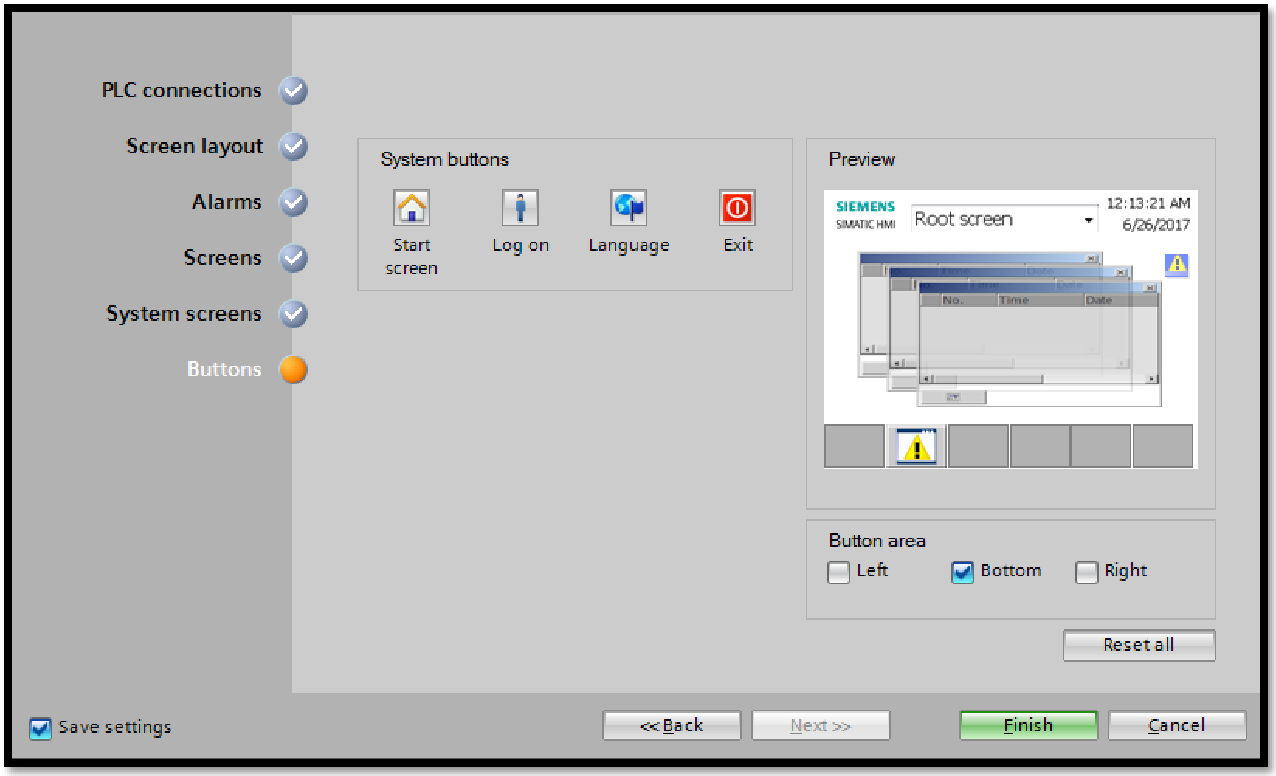
☑ Language switching

☑ Stop Runtime

☑ Select all

Buttons: These are shortcuts that allow you to access essential functions

You can customize the order of the buttons by dragging them to the bottom sections of the Preview window. It is suggested to have Start screen and Exit as a default. Click Finish when done.

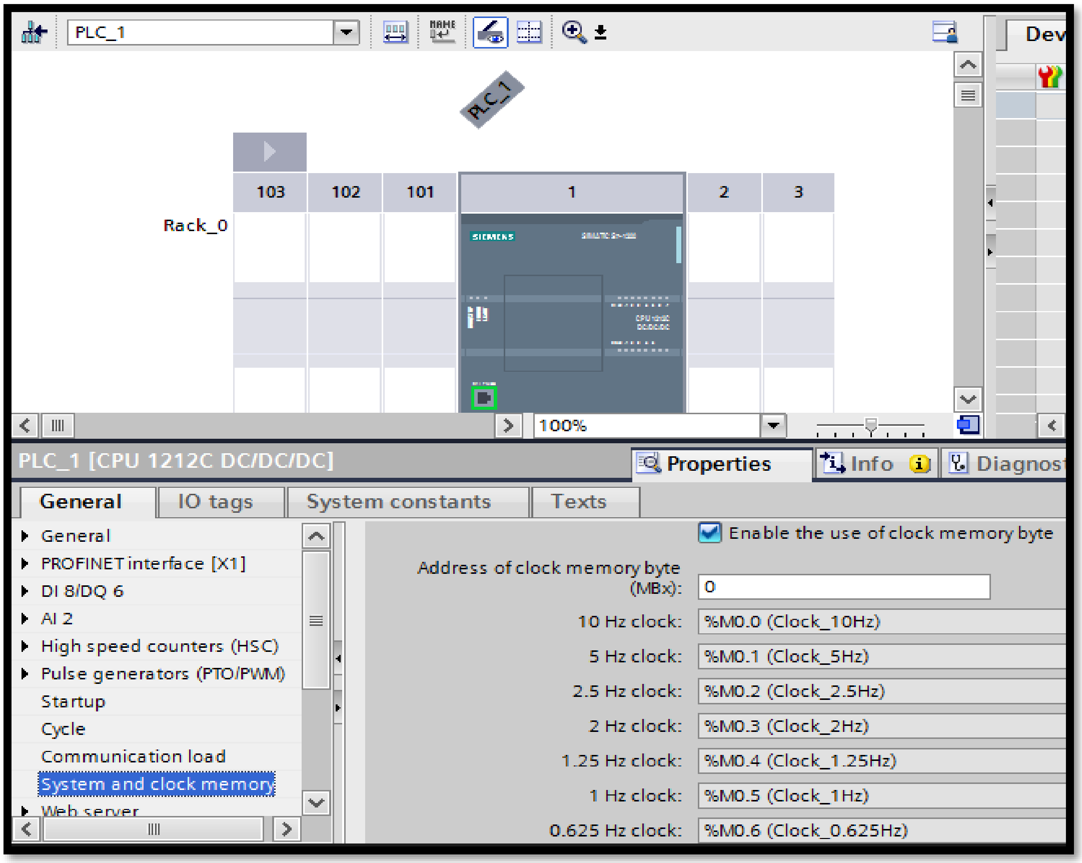


Button area
□ Left ☑ Bottom □ Right

☑ Save settings

Configuring and Practicing:

Go to PLC_1 then into Device Configuration. From there, open the Properties option then enter Systems and clock memory. Enable to Clock memory bits.



☑ Enable the use of clock memory byte

Address of clock memory byte (MBx): 0

10 Hz clock: %M0.0 (Clock_10Hz)

5 Hz clock: %M0.1 (Clock_5Hz)

2.5 Hz clock: %M0.2 (Clock_2.5Hz)

2 Hz clock: %M0.3 (Clock_2Hz)

1.25 Hz clock: %M0.4 (Clock_1.25Hz)

1 Hz clock: %M0.5 (Clock_1Hz)

0.625 Hz clock: %M0.6 (Clock_0.625Hz)

Download to your PLC

Now that you have successfully configured your HMI screen, go to the root screen by going to the project tree on the left and opening the HMI drop down menu. Double click root screen to open.

At the root screen, you can see all the different tabs for your sub screens.

There is a tool box on the right hand side for you to work with.

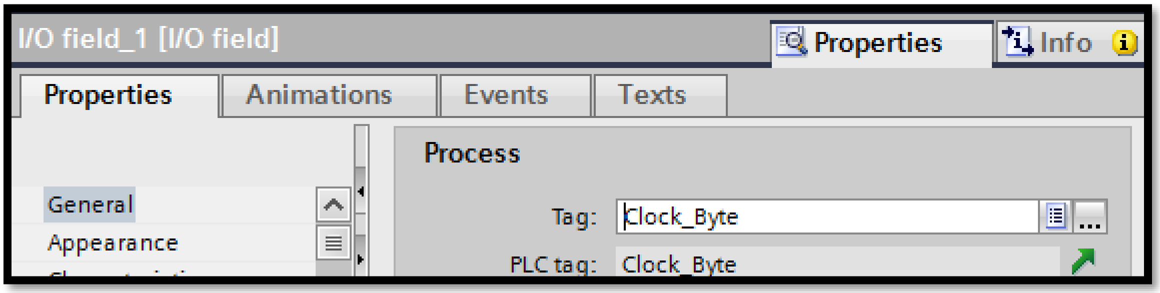
Basic objects

Elements

Controls

Now to test to see if your screen is linked to your PLC, we will create a clock byte on Root screen:

1. Drag an I/O field from Elements to the root screen.
2. In the properties of the I/O field click text format and choose Center for text.
3. Also the properties of I/O field, click PLC tag default table, then assign “Clock_Byte” as the address.



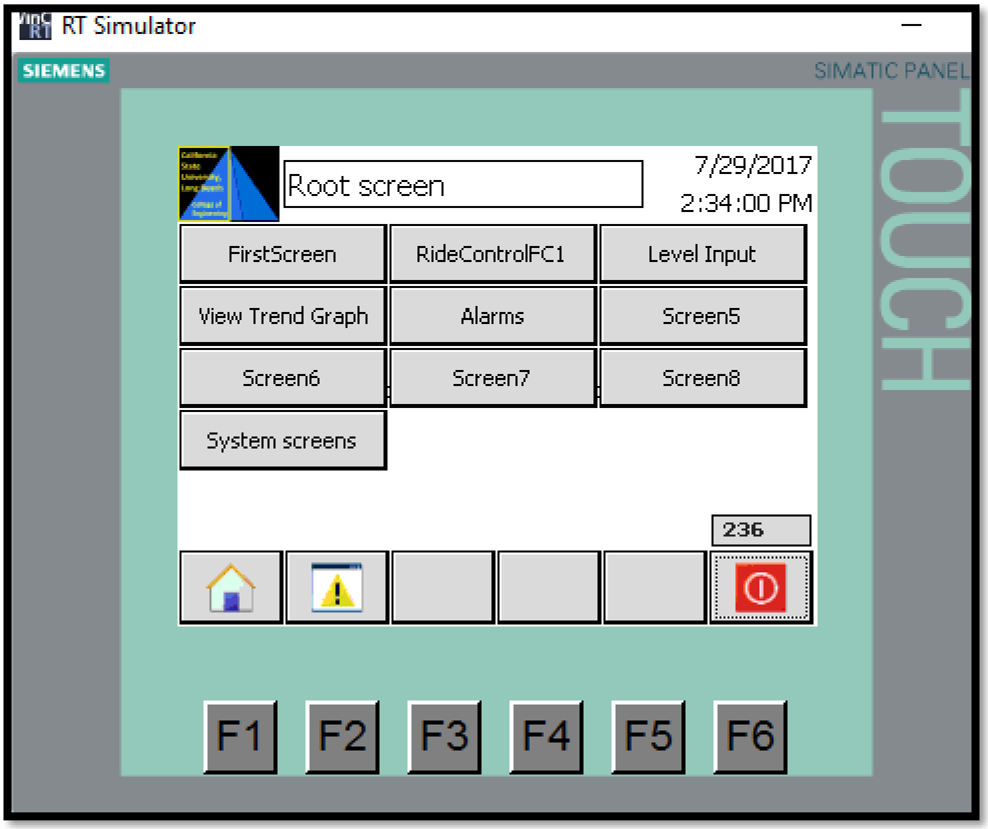
I/O field_1 [I/O field]

Process

Tag: Clock_Byte

PLC tag: Clock_Byte

4. The clock bytes indicate the communication between HMI and PLC
5. Before starting the simulator for the HMI screen, you will first need to download to your PLC and start the simulator for the PLC
6. Once done, you can now open the HMI simulator
7. If done correctly, the I/O field should be changing in real time



8. Save the file

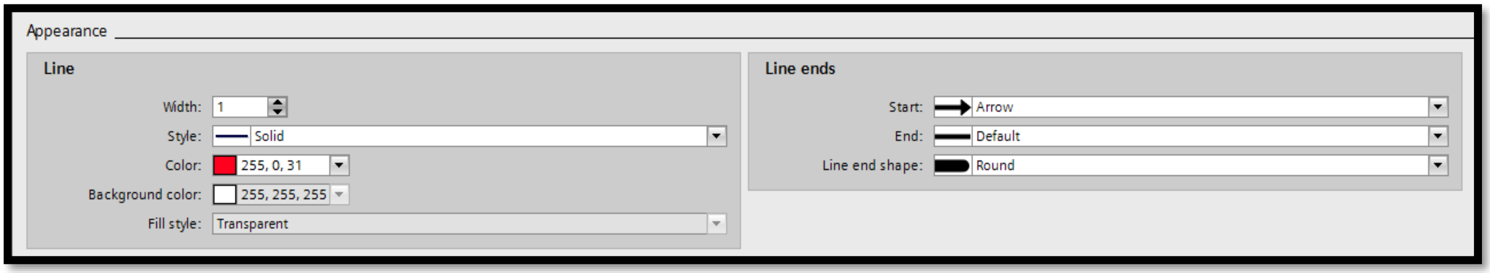
Lab 7: Operating Human Machine Interface Part 2

Since we have the HMI configured and communicating with the PLC, now we will start playing with different objects in the toolbox to get familiar with them.

Practicing with different objects:

We will start off by going entering Screen1 from the Project Tree. This will be under Screens in the HMI drop down menu.

1. First we will need to rename the screen from “Screen1” to “FirstScreen”. We can do this by right clicking on the screen name in the menu and selecting rename.
2. After that is done, begin playing with the objects from toolbox menu on the right.
3. Grab Line object and drag it across the screen. Enter the object’s properties and examine the Appearance option. Play around with the options (ie. Color, width, style, arrow).



Appearance

Line

Width: 1

Style: Solid

Color: 🟥 255, 0, 31

Background color: □ 255, 255, 255

Fill style: Transparent

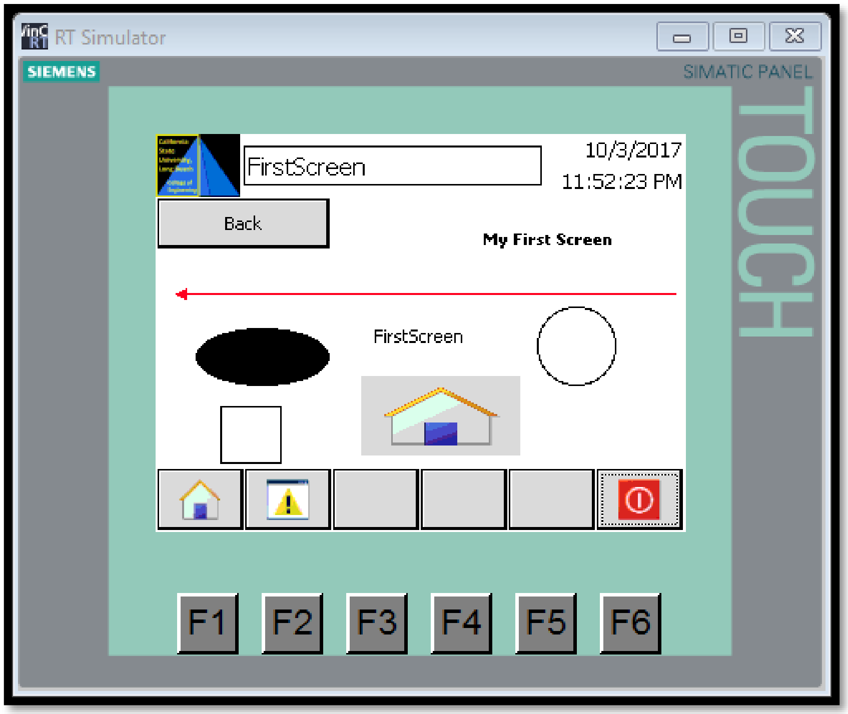
Line ends

Start: Arrow

End: Default

Line end shape: Round

4. Now grab the text field and add it to the screen. Here, we will type “My First Screen”. Enter the objects properties and change the text color to black. Examine the rest of the options and play with it.
5. Grab the various shapes and place it on the screen. Play with the size and properties.
6. Finally, grab the Graphic View and place it on the screen. Here, we can enter the properties and select what graphics we want in the field. The option at the bottom (left) in General allows us to upload our own image.
7. Click save and begin the simulation. Observe the FirstScreen.



Here is how the simulator should look like.

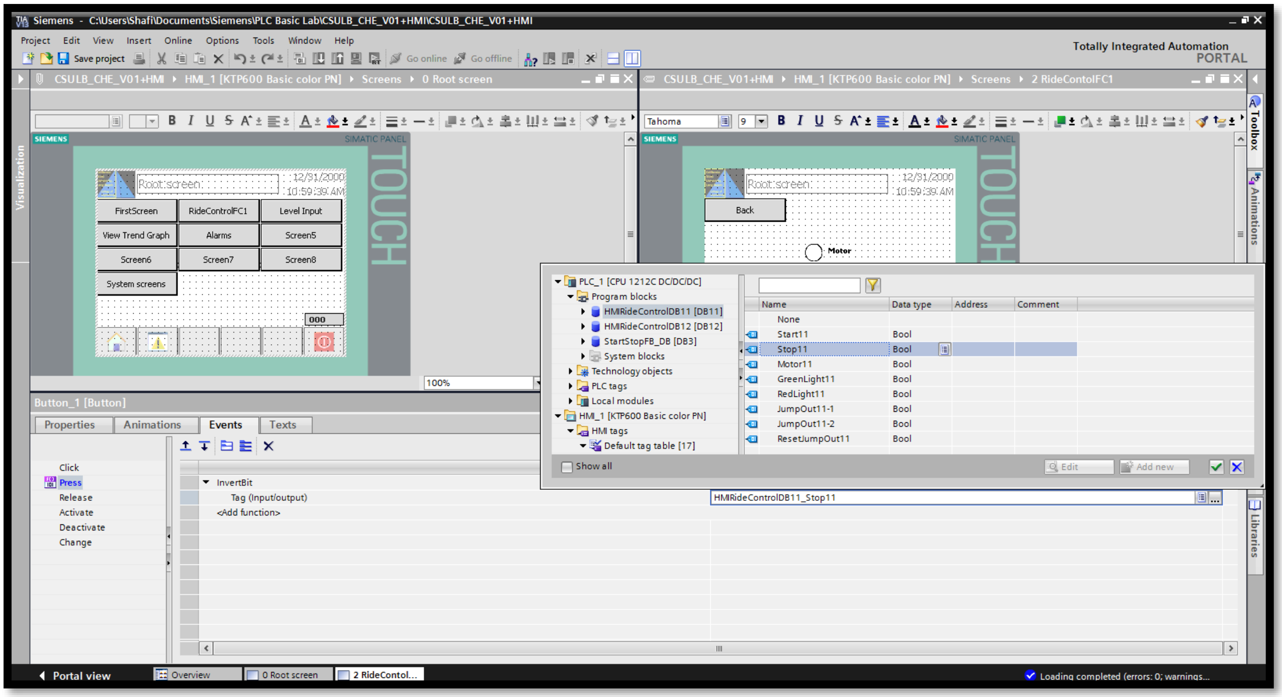
Now that we got screen 1 configured, we will now practice with screen 2. We will link this with the RideControlFC1 (Lab 2) and have flashing lights and controllable buttons.

Note: Do not use real world addresses.

Linking the HMI with the PLC programs:

1. Highlight Screen 2 and rename it RideControlFC1
2. Once done, enter the screen by double clicking it
3. From here, we will add 2 buttons and 3 circles to act as lights
4. Rename one of the buttons to “Stop Ride” and the other to “Start Ride” by double clicking them. We will need to add an I/O field under each of the button.
5. Now that’s set, we will link the buttons to the RideControlFC1 tags. This is done by entering the properties of the buttons. Select Stop Ride button and enter its properties.

6. Under the Events Tab, enter the Press option and add function “Invert Bit”. Link this function to “HMIRideControl_Stop11” tag. Do the same thing for the Release option.



Button_1 [Button]

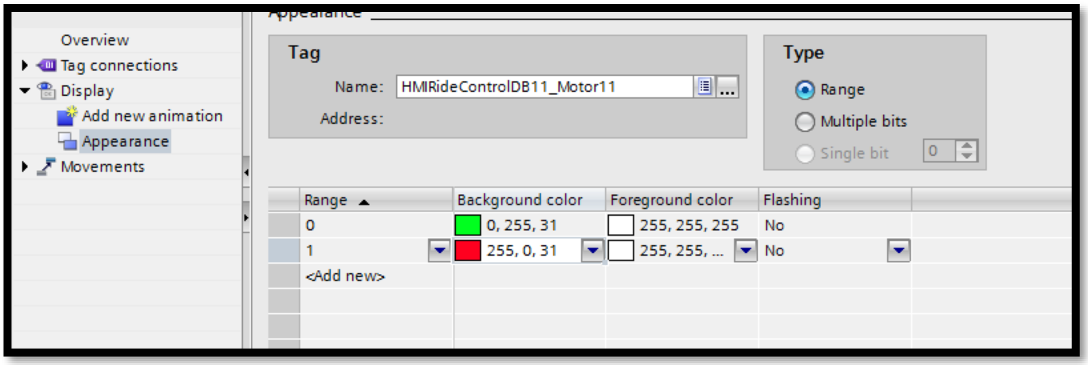
Properties | Animations | Events | Texts

Click
Press
Release
Activate
Deactivate
Change

InvertBit
Tag (Input/output) HMIRideControlDB11_Stop11
<Add function>

Name	Data type	Address	Comment
None			
Start11	Bool		
Stop11	Bool		
Motor11	Bool		
GreenLight11	Bool		
RedLight11	Bool		
JumpOut11-1	Bool		
JumpOut11-2	Bool		
ResetJumpOut11	Bool		

7. Now under animation, make the button turn red when its 1 and green when its 0. Tag this color change to the Motor.



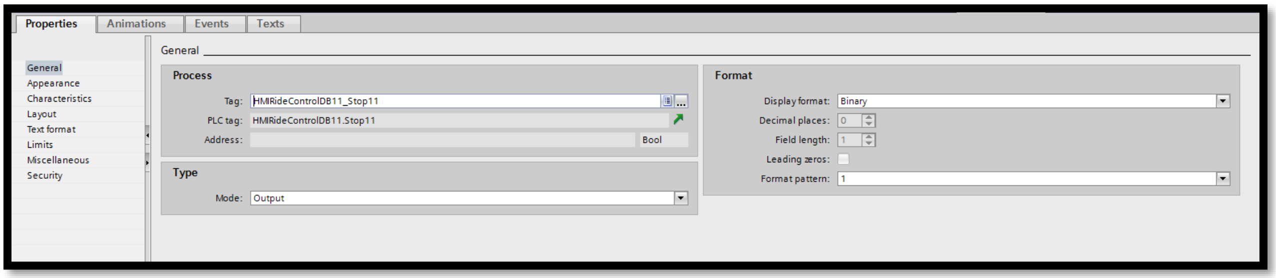
Overview
Tag connections
Display
Add new animation
Appearance
Movements

Tag
Name: HMIRideControlDB11_Motor11
Address:

Type
● Range
○ Multiple bits
○ Single bit 0

Range	Background color	Foreground color	Flashing
0	🟩 0, 255, 31	□ 255, 255, 255	No
1	🟥 255, 0, 31	□ 255, 255, ...	No
<Add new>			

8. Repeat the previous steps with Start Ride button. But invert the color to 0 as red and green as 1.
9. Now enter the I/O field properties under the Stop Ride button. Here we will link this field with the same tag as the button respectfully. This is done under the General Tab and display format set to Binary. Set Mode to Output only.



General

Process

Tag: HMIRideControlDB11_Stop11

PLC tag: HMIRideControlDB11.Stop11

Address: Bool

Type

Mode: Output

Format

Display format: Binary

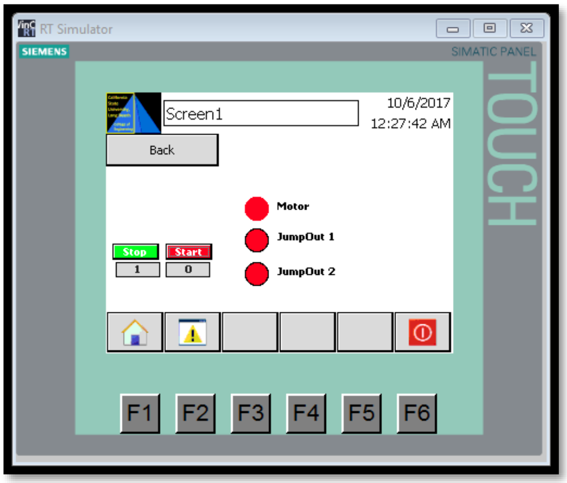
Decimal places: 0

Field length: 1

Leading zeros: □

Format pattern: 1

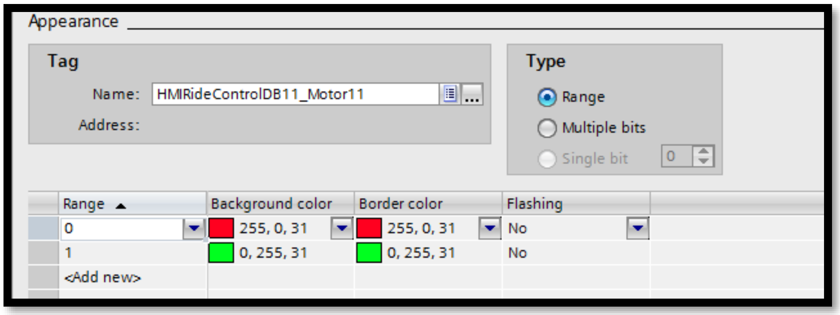
10. Repeat with the Start Ride I/O field.
11. Download and start simulator and test the buttons. The I/O should show a “0” when pressed.



Now since we got the buttons linked, we can get the circles to flash.

Circles as Light:

1. On the same screen, place a text field next to each circle and rename it “Motor”, “JumpOut 1”, and “JumpOut 2” respectfully.
2. Enter the properties of the Motor circle and enter the apperence option. Just like we did for the button display, we will set bit 1 to show green and 0 to red.
3. Link this to “HMIRideControlDB11_Motor11”.



Appearance

Tag

Name: HMIRideControlDB11_Motor11

Address:

Type

● Range

○ Multiple bits

○ Single bit 0

Range	Background color	Border color	Flashing
0	🟥 255, 0, 31	🟥 255, 0, 31	No
1	🟩 0, 255, 31	🟩 0, 255, 31	No
<Add new>			

4. Repeat this for the other 2 circles but link the circle to its respected tags.
5. Download and save. Start simulation and test.

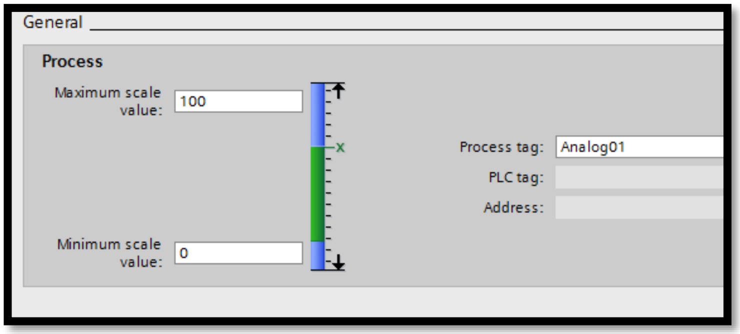
Lab 8: Operating Human Machine Interface Part 3

Till now, we have configured 2 screens, one for practice and another linking to the PLC. Now we will play with indicators, inputs, and graphs.

For this lesson, we will link the objects to internal tags instead of real Analog inputs.

Making a level indicator linked with an I/O field:

1. First we will start by adding a new tag in the HMI. Under HMI tags, add a tag and name it Analog01. Set the data type to INT.
2. Now go to screen 3 and rename it Level Input.
3. Find the Bar object and also an I/O field and place it on the screen.
4. Enter the Bar object and under general tag, set the tag to Analog01. Here we will set the max to 100 and min to 0.



General

Process

Maximum scale value: 100

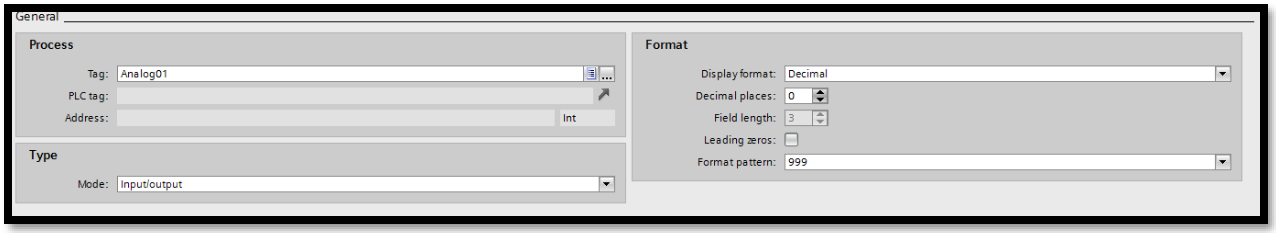
Minimum scale value: 0

Process tag: Analog01

PLC tag:

Address:

5. Now enter the I/O field properties and set the tag to Analog01. Mode should be set to Input/Output, Display to decimal, and Format pattern to “999”.



General

Process

Tag: Analog01

PLC tag:

Address: Int

Type

Mode: Input/output

Format

Display format: Decimal

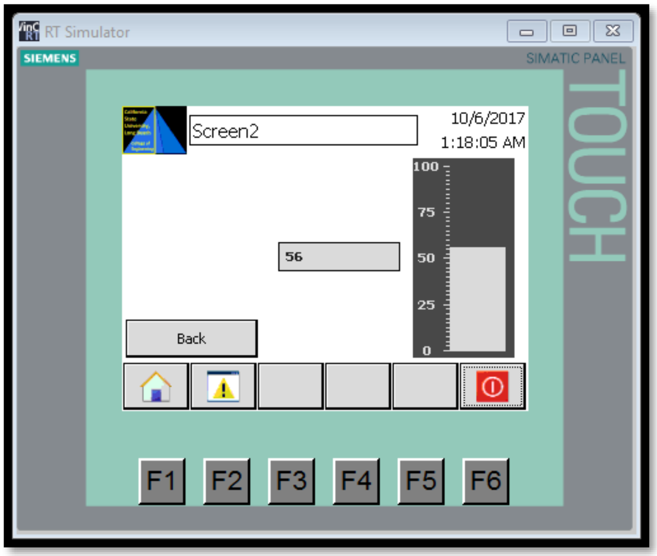
Decimal places: 0

Field length: 3

Leading zeros: □

Format pattern: 999

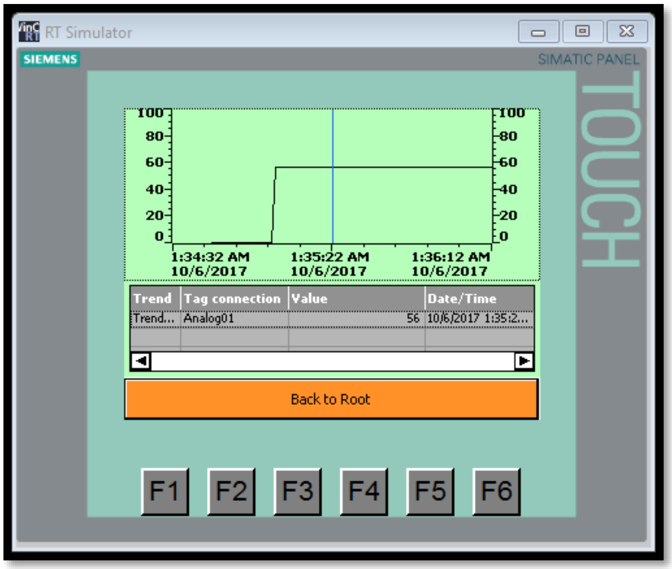
6. Download and test. We should be able to input a value into the I/O field from the simulator when we click it.



Next we will work with the trend graph to see how the data change in the level indicator will affect the trend.

Trend Graph:

1. Go to screen 4 and rename it View Trend Graph.
2. Find the Trend View object and place it on the screen.
3. Enter its properties and add a new trend. Configure Source settings to Analog01. Leave everything else as is.
4. In the Right/Left value axis option on the left, set axis start to 0 and end to 100. In Time axis, set interval to 100 and axis mode to time.
5. Download and start simulation. Go into the Level Input screen first and enter a value. Then go into the Trend Graph screen and observe the graph change.
6. Save the project.



Trend	Tag connection	Value	Date/Time
Trend...	Analog01	56	10/6/2017 1:35:2...

Lab 9: Operating Human Machine Interface Part 4

The final lab to this class will be about Alarms. This is a very important part for a manufacturing facility. The alarm screen indicates if there is any errors within the system, such as a broken connection, malfunctioned equipment, or if a unit has reached it max or min. This allows the operator to troubleshoot the issue with ease knowing the general direction to look at.

Alarm:

1. Enter the PLC tag and add a new tag, Name it “AlarmWord” and set the data type to Word. Set the address to “%MW60”
2. Now go to the HMI tag and add a new tag named “AlarmWord” as well. Connection set to HMI_Connection_1 and PLC tag set to “AlarmWord”.
3. Under Discrete alarms, set an example alarm with name Tank Level High. Alarm class is set to Errors and Trigger Bit to 0.
4. Now go to screen 5 and rename it Alarms.
5. Find the Alarm View and insert it into the screen. Also grab an I/O field and place it under the alarm window.
6. For the I/O field, set the tag to AlarmWord with Mode to Output.
7. Display format is set to decimal and format pattern to “9999999999999999”.
8. In the alarm window under General option, check Errors, Warnings, and System. In Columns option, check all visible columns except for Acknowledgement group, Diagnosable, and Devices.
9. Save the project.