

TouchPAD

TPD/VPD Series HMI Devices Programming Guide

Version: 1.0.6 June 2020



Usage information for

TPD-2.8 inch series

TPD-4.3 inch series

TPD-7.0 inch series

VPD-3.5 inch series

VPD-4.3 inch series

VPD-7.0 inch series



Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright @ 2020 by ICP DAS CO., LTD. All rights are reserved.

Trademarks

Names used for identification only may be registered trademarks of their respective companies.

Contact US

If you have any questions, please feel free to contact us.
We will give you a response within about two working days.

Email: service@icpdas.com

Table of Contents

Chapter 1. Introduction -----	5
1.1 Hardware Specifications -----	6
Chapter 2. Software Installation -----	7
2.1 Install the HMIWorks Software-----	7
2.2 Create a New Project-----	9
2.3 HMIWorks Working Environment-----	11
2.4 Downloading Methods for TouchPAD -----	12
Applications are downloaded through RS-485. -----	12
Applications are downloaded through Ethernet. -----	14
Applications are downloaded through USB. -----	17
Chapter 3. Demo Programs-----	21
Chapter 3. -----	21
3.1 Using the standard C language -----	21
3.1.1 Demo 1 (Beep, Timer, Counter) -----	21
3.1.2 Demo 2 (Use Slider to control the object) -----	28
3.1.3 Demo 3 (Digital Clock) -----	33
3.1.4 Demo 4 (Remotely control the ICPDAS I/O model) -----	39
3.2 Using the Ladder Designer -----	47
3.2.1 Demo 1 (Beep & Count)-----	47
3.2.2 Demo 2 (Displaying the Decimal Point)-----	56
3.2.3 Demo 3 (Digital Clock) -----	59
3.2.4 Demo 4 (Remotely control the ICPDAS I/O model) -----	66
3.3 Demo List-----	73

Appendix A. Downloading Information -----	74
Appendix B. Creating Your Own GUI -----	75
Revision History -----	76

Chapter 1. Introduction

The TPD series and VPD series devices are part of the TouchPAD series that includes a tiny touch HMI interface and are designed for building and home automation. The TouchPAD series is equipped with a high resolution TFT color touch screen and can be installed using a regular electrical wall-mount outlet. The TouchPAD series can also be seamlessly integrated with rich I/O modules and provides a beautiful, flexible and customized screen display. In short, the TouchPAD series is the best choice when upgrading from the mechanical switches to intelligent control pads.

For PLC users, the HMIWorks utility includes Ladder Designer, and for IT users, a C language environment is provided. It takes less than 30 minutes to learn how to create an application for a TouchPAD device when using Ladder Designer. With all the features provided, TouchPAD touch HMI devices are possibly the most cost-effective HMI device solution that has been created.

1.1 Hardware Specifications

If you are interested in learning more about the hardware specifications for the various TouchPAD models, please refer to our website:

http://www.icpdas.com/en/product/guide+Panel_Products+TouchPAD+TPD_Series



TPD Series
(2.8" Touch Screen)



TPD Series
(4.3" Touch Screen)



TPD Series
(7.0" Touch Screen)



VPD Series
(3.5" Touch Screen)



VPD Series
(4.3" Touch Screen)



VPD Series
(7.0" Touch Screen)

Chapter 2. Software Installation

First, install the HMIWorks development software on your PC. To download the latest version, visit the website at:

<http://www.icpdas.com/en/download/show.php?num=944&nation=US&kind1=&model=&kw=HMIWORKS>

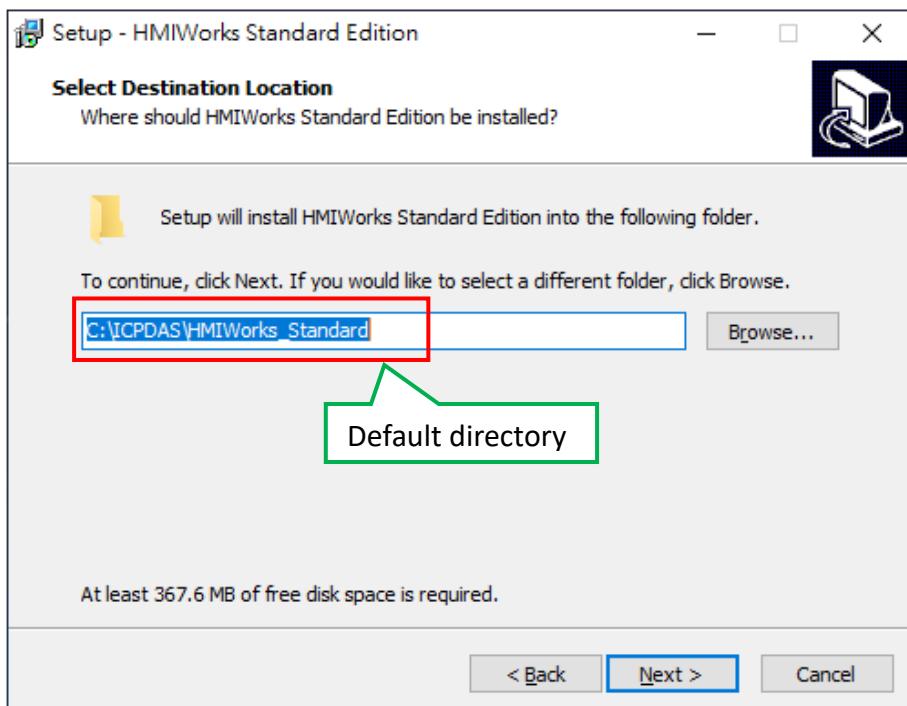
2.1 Install the HMIWorks Software

Step 1: Double click the execution file, i.e., HMIWorks_STD_vxxx_setup.exe, to install the HMIWorks software.

(This manual uses V2.10.02)



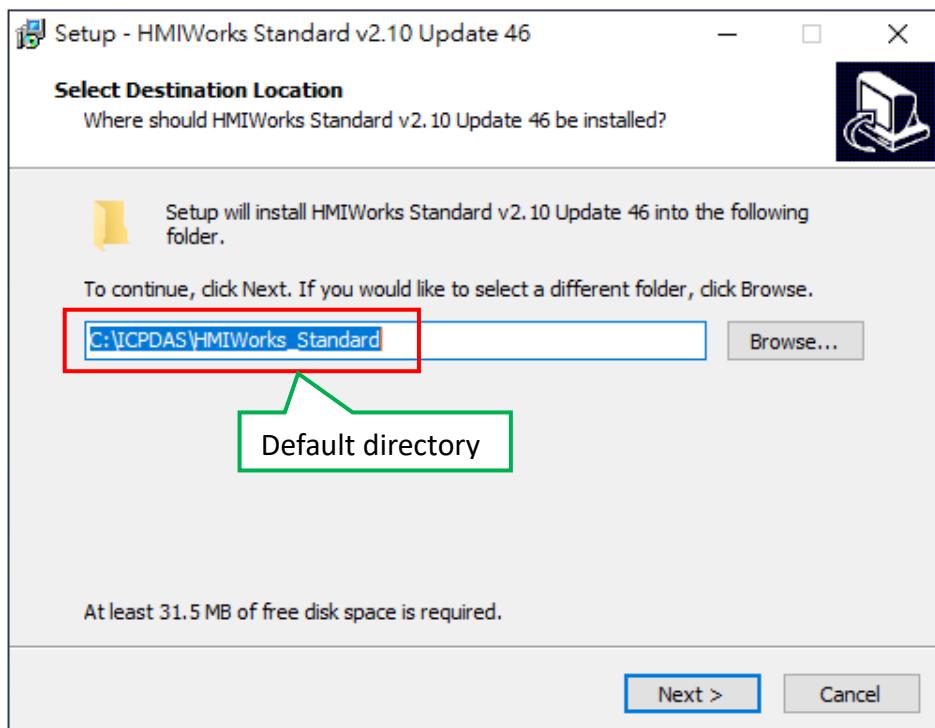
Step 2: Follow the instructions in the setup wizard and click “Finish” to complete the installation.



Step 3: Double click the execution file, i.e., HMIWorks_STD_vXXX_Update_XX.exe, to update the HMIWorks software.
(This manual uses V2.10.46)

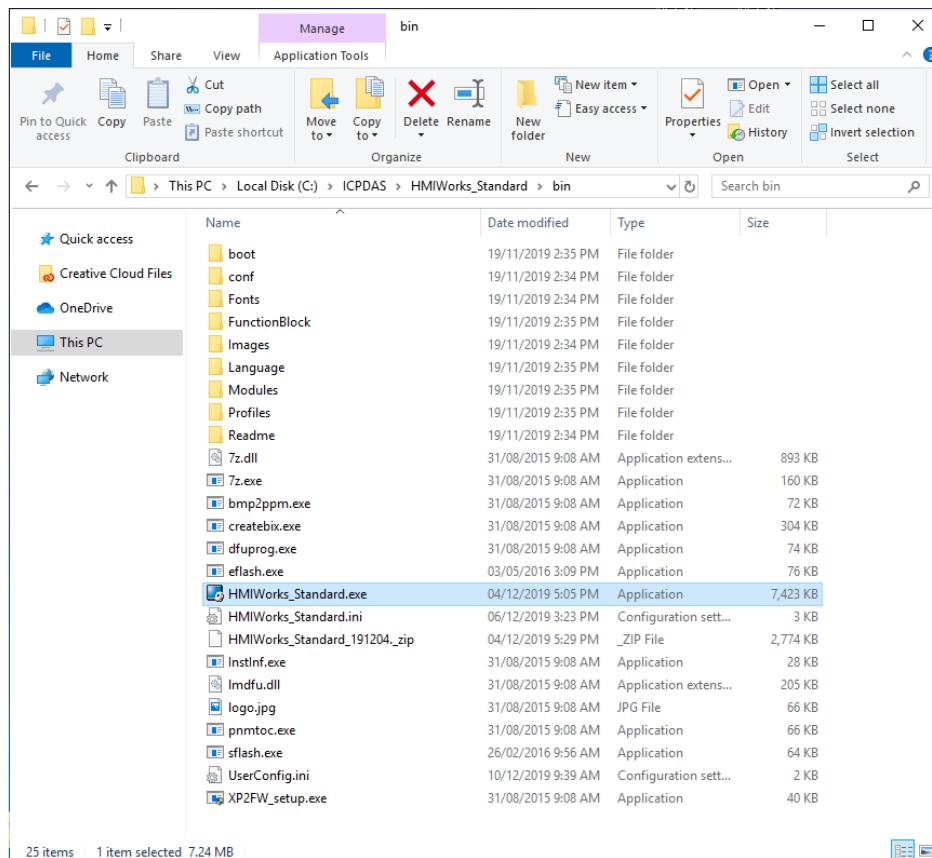


Step 4: Follow the instructions in the update wizard and click “Finish” to complete the update.

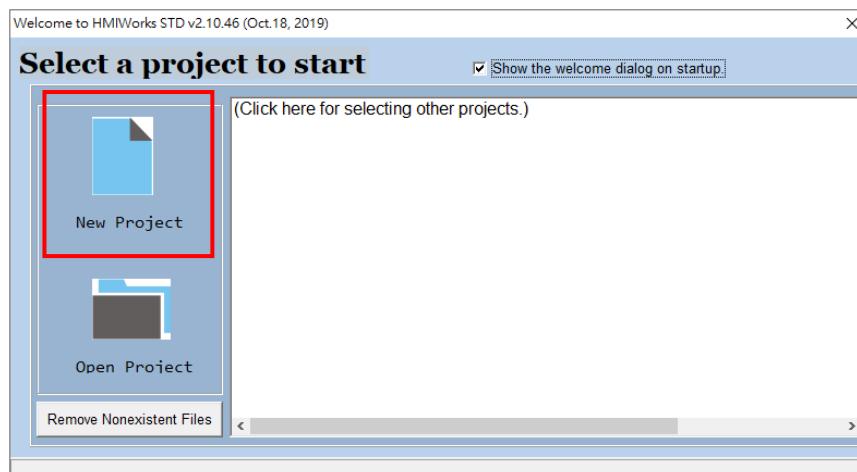


2.2 Create a New Project

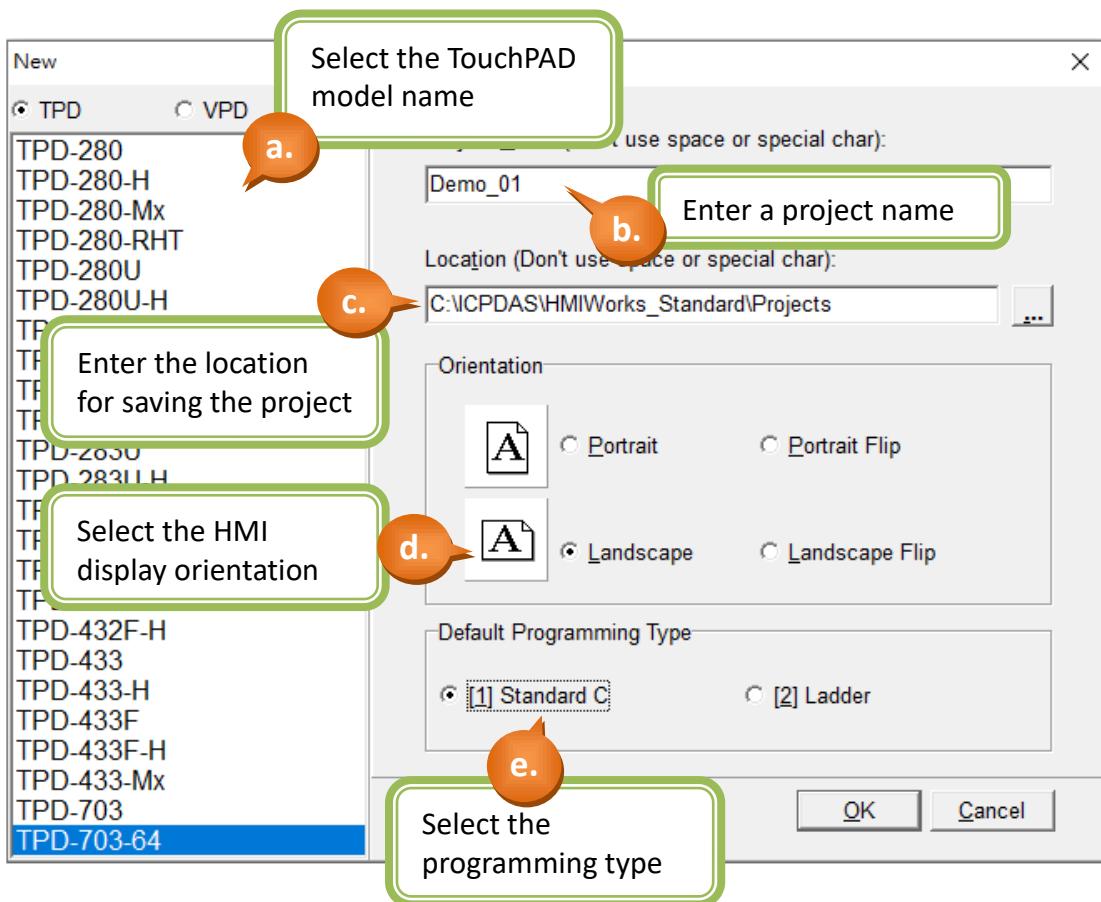
Step 1: After the HMIWorks software has been successfully installed, execute the software “HMIWorks_Standard.exe” from the installed folder.
(Default : C:\ICPDAS\HMIWorks_Standard\bin)



Step 2: Select the “New Project” icon to create a new project.

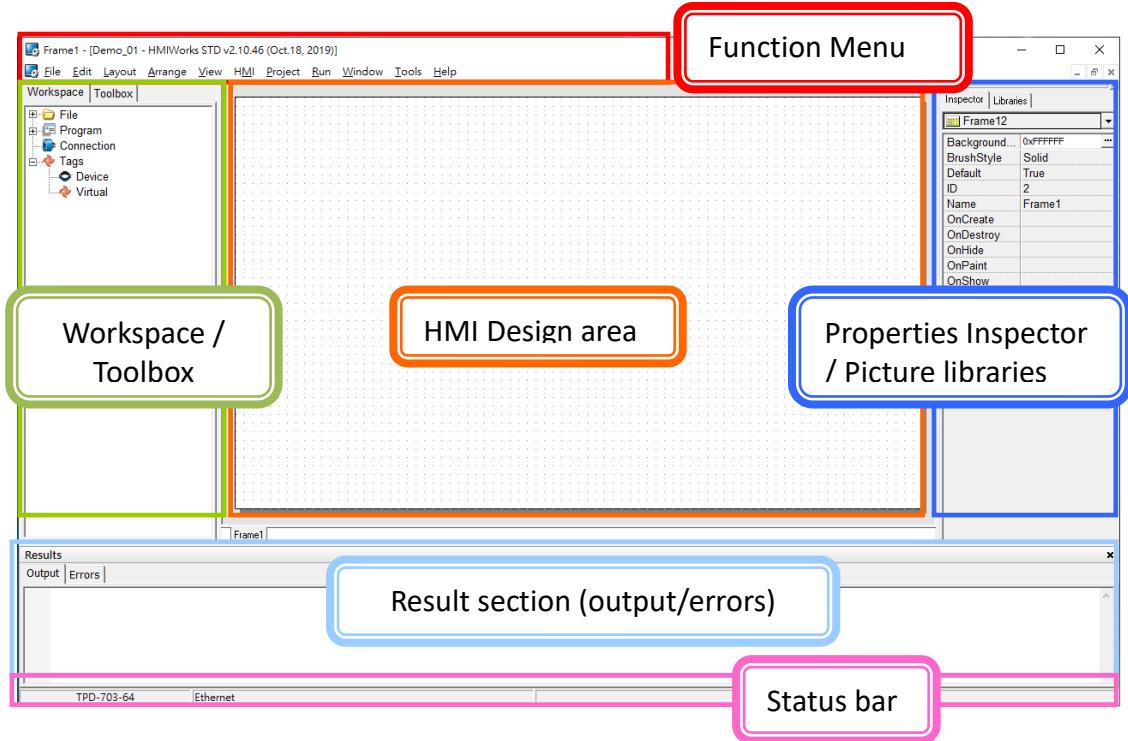


Step 3: Select the TouchPAD model name (a), enter a project name (b) and its location where the project should be saved (c), then select the display orientation (d) and the default programming type (e), and click "OK".

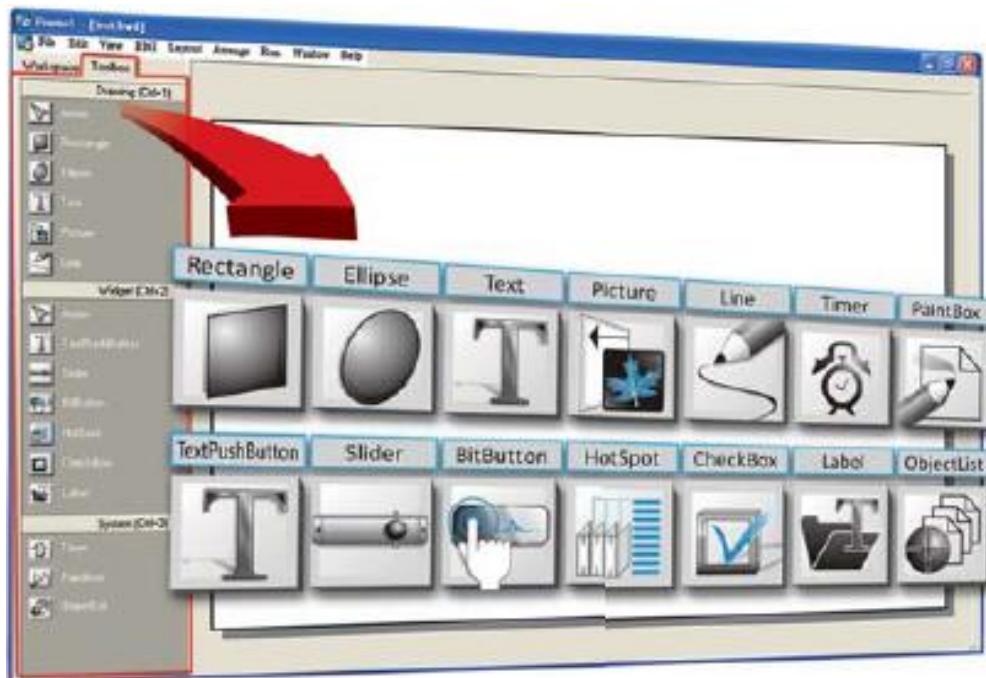


NOTE: If the TouchPAD model selected is incorrect, it will cause errors in the program and will mean that the project cannot be downloaded.

2.3 HMIWorks Working Environment



Toolbox:



2.4 Downloading Methods for TouchPAD

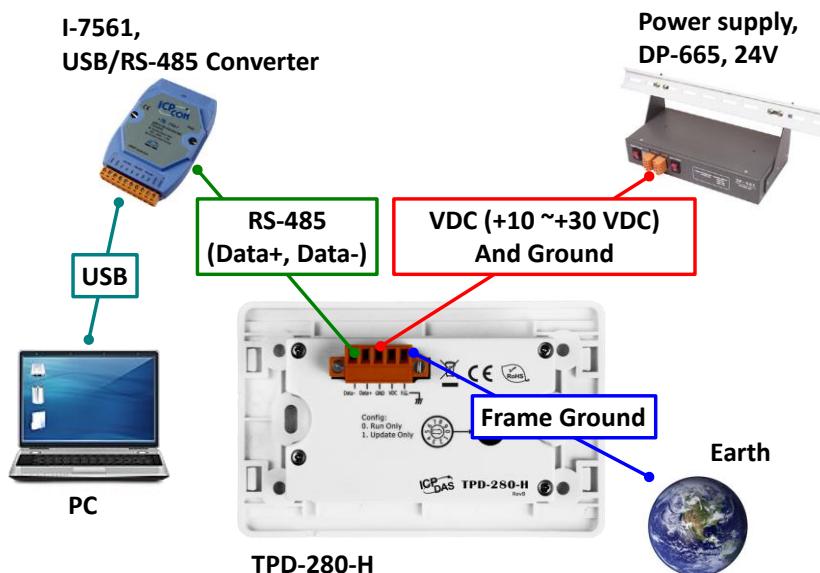
Applications are downloaded through **RS-485**.

Applicable Model:

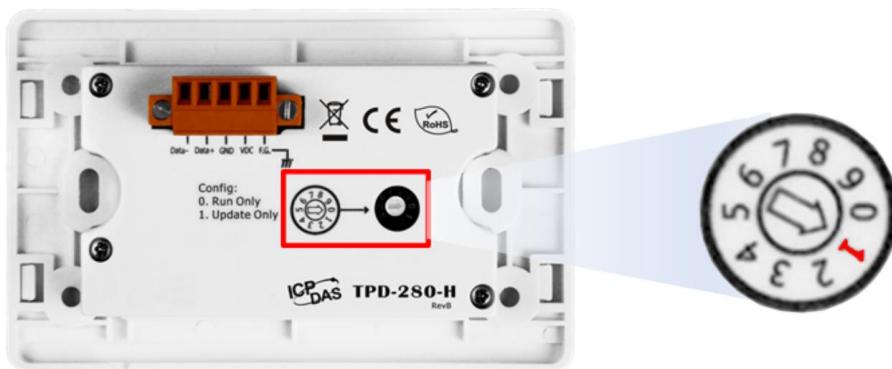
TPD-280-H TPD-280-M1 / M2 / M3

Step 1: Connect the TouchPAD model and the computer using the RS-485 converter.

For example : The ICPDAS product I-7520 (RS-232 / RS-485 Converter) or I-7561 (USB / RS-485 Converter)



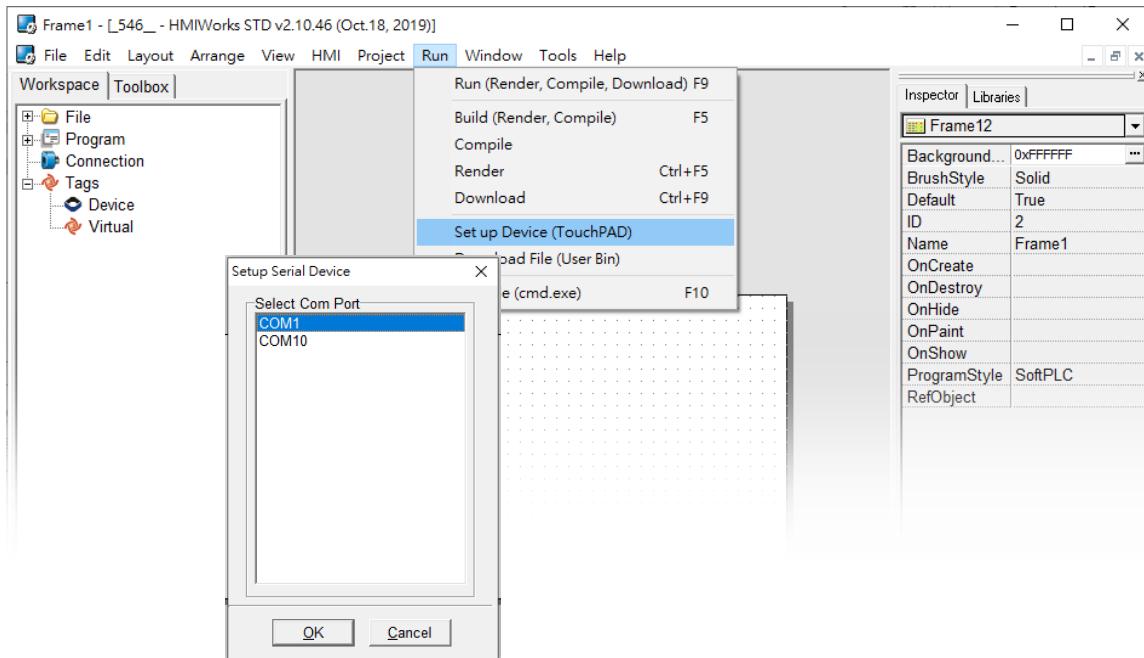
Step 2: Adjust the Rotary Switch to “1” (Update mode) and then power on the TouchPAD model. After powering on the TouchPAD model, the screen of the TouchPAD should be blank.



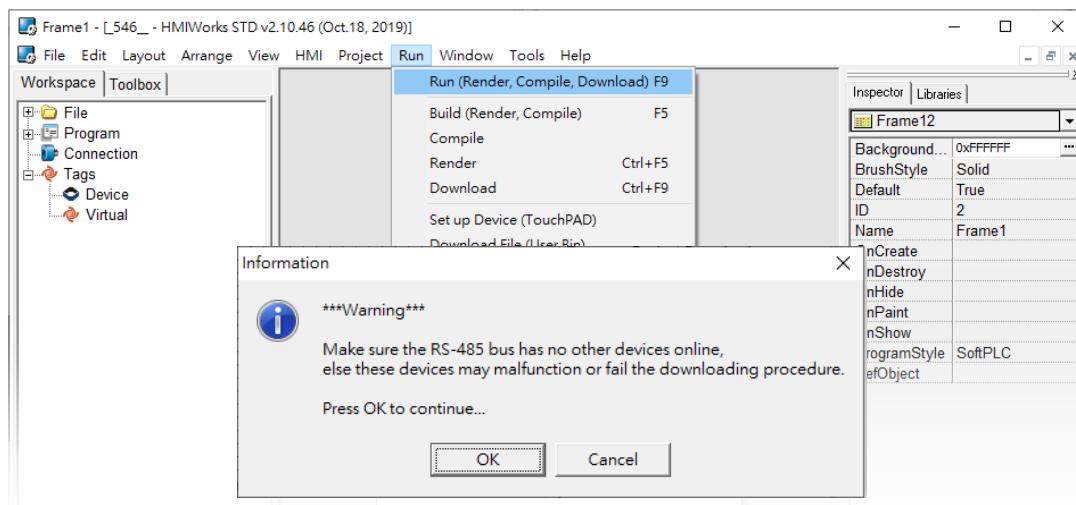


NOTE : The default setting of Rotary Switch on TouchPAD is “0” (Run mode)

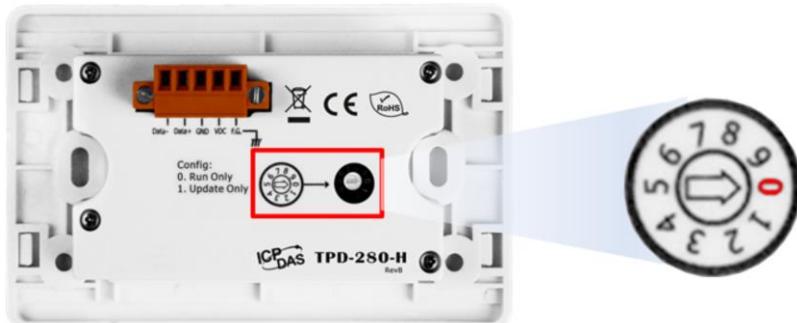
Step 3: Select the “Set up Device (TouchPAD)” item from the “Run” menu to open the “Setup Serial Device” dialog box. Select the COM Port that connects the TouchPAD.



Step 4: In the HMIWorks application, click the “Run (Render, Compile, Download) F9” item from the “Run” menu, or press F9. After completing the building, the reminded-window will bounce (**Please make sure there are no other device on the RS-485 bus**), click the “OK” button to start downloading.



Step 5: Wait until the Download are completed(100%). Adjust the Rotary Switch to "0" (Run mode), and then reboot the TouchPAD. After rebooting the TouchPAD, the screen of the TouchPAD will display the project.

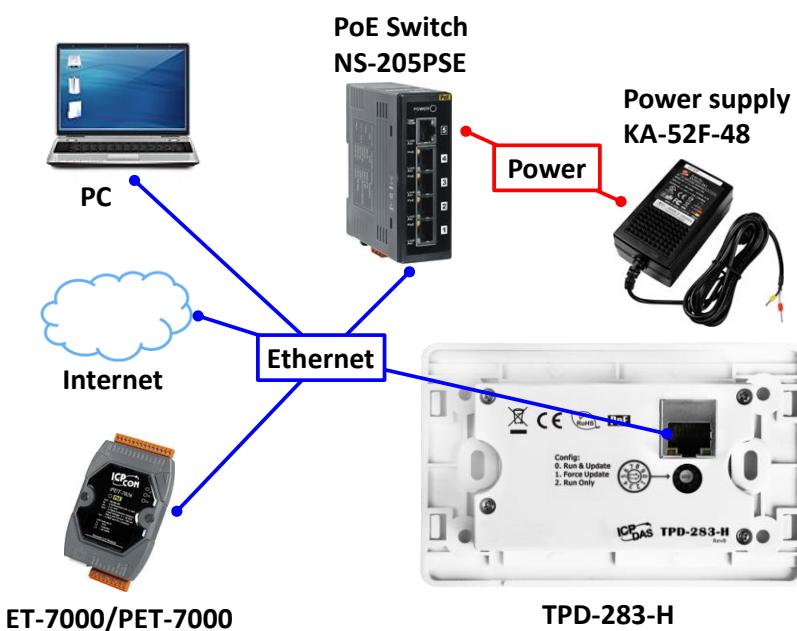


Applications are downloaded through **Ethernet**.

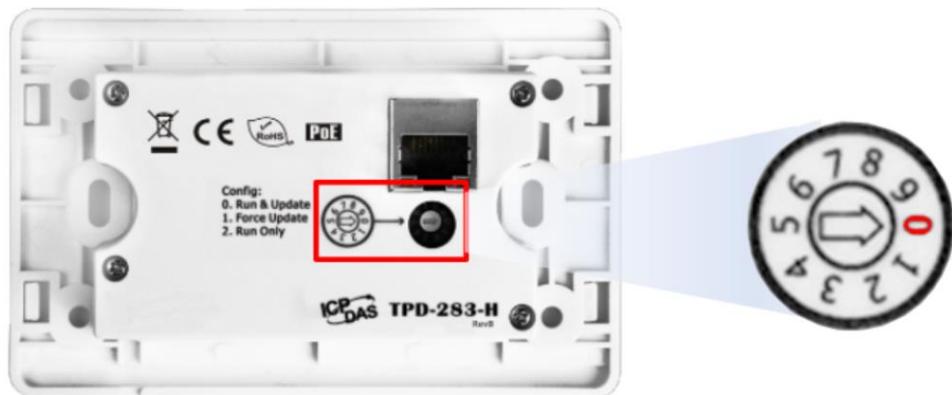
Applicable Model:

- | | |
|--|---|
| <input checked="" type="checkbox"/> TPD-283(U)-H | <input checked="" type="checkbox"/> TPD-283(U)-M1 / M2 / M3 |
| <input checked="" type="checkbox"/> TPD-433-H(-EU) | <input checked="" type="checkbox"/> TPD-433F-H |
| <input checked="" type="checkbox"/> TPD-433F-M2 | <input checked="" type="checkbox"/> TPD-703(-64) |
| <input checked="" type="checkbox"/> VPD-133(N)-H | <input checked="" type="checkbox"/> VPD-143(N)-H |
| <input checked="" type="checkbox"/> VPD-173N(-64) | <input checked="" type="checkbox"/> VPD-173X(-64) |

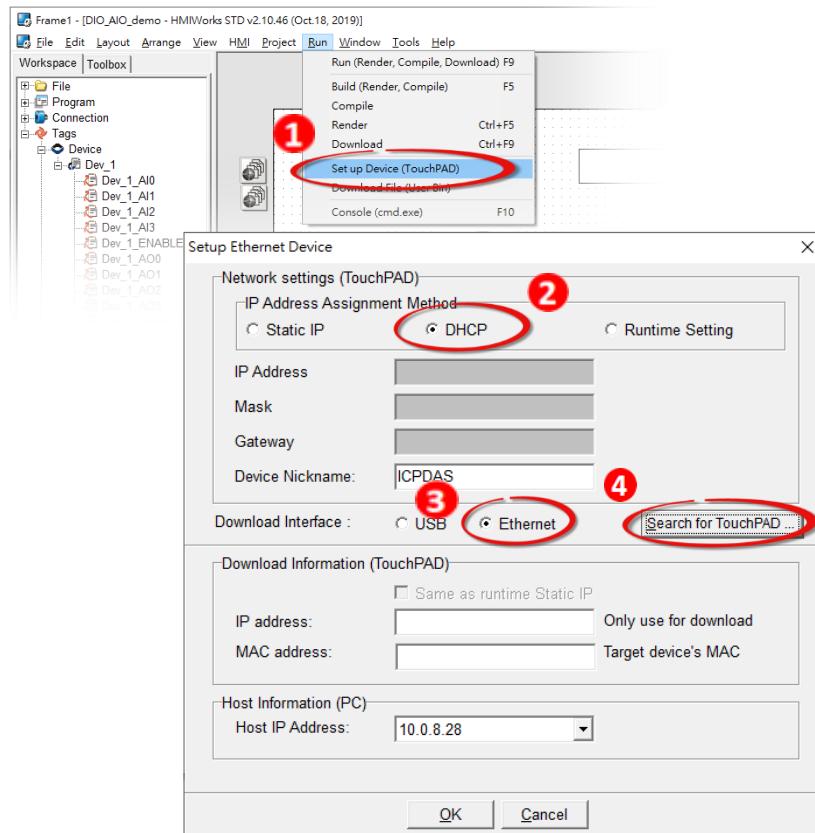
Step 1: Connect both the TouchPAD model and the Host PC to the same sub network, or use a Power over Ethernet Switch, e.g., an NS-205PSE, and supply power to the TouchPAD model via the PoE Switch.



Step 2: Adjust the Rotary Switch to “0” (Run & Update mode) and then reboot the TouchPAD model.

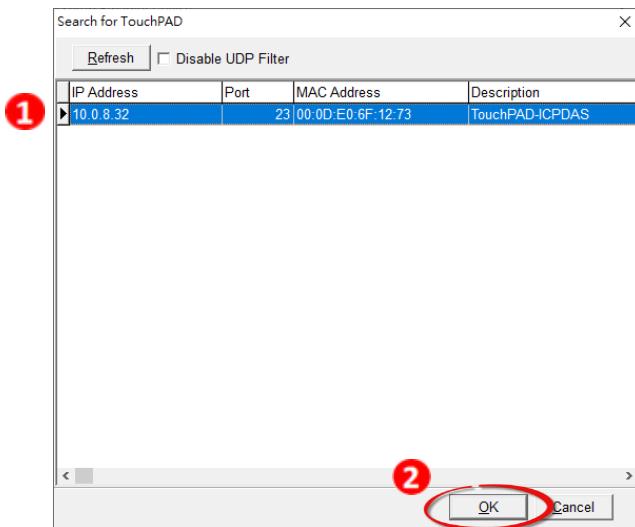


Step 3: Select the “Set up Device (TouchPAD)” item from the “Run” menu to open the “Setup Ethernet Device” dialog box. Select the “DHCP” and “Ethernet” options, and then click the “Search for TouchPAD...” to search for the model



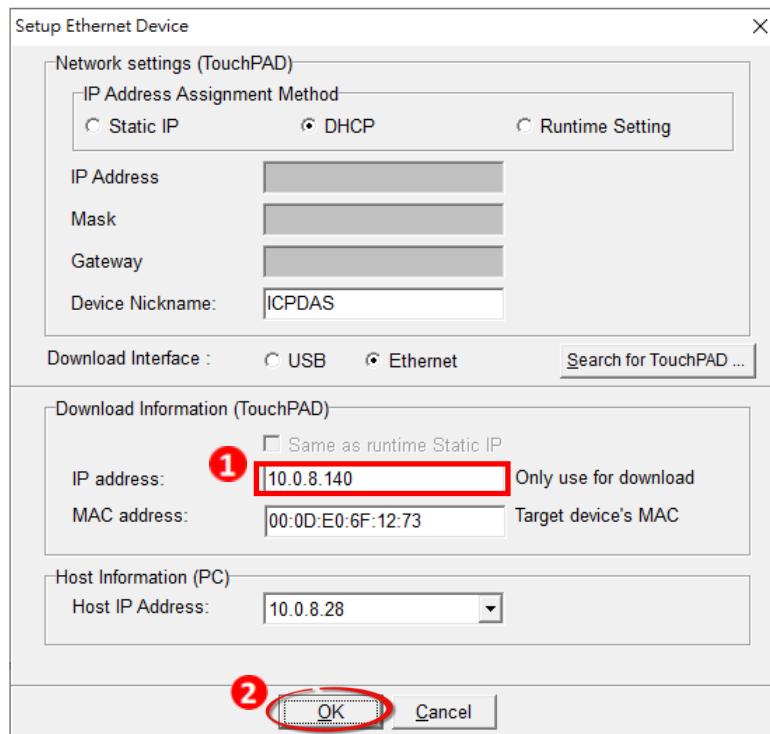
NOTE : If your network doesn't have access to a DHCP, you can use the Static IP and enter the effective IP information.

Step 4: Select the TouchPAD model that has the correct MAC address, and then click the “OK” button.

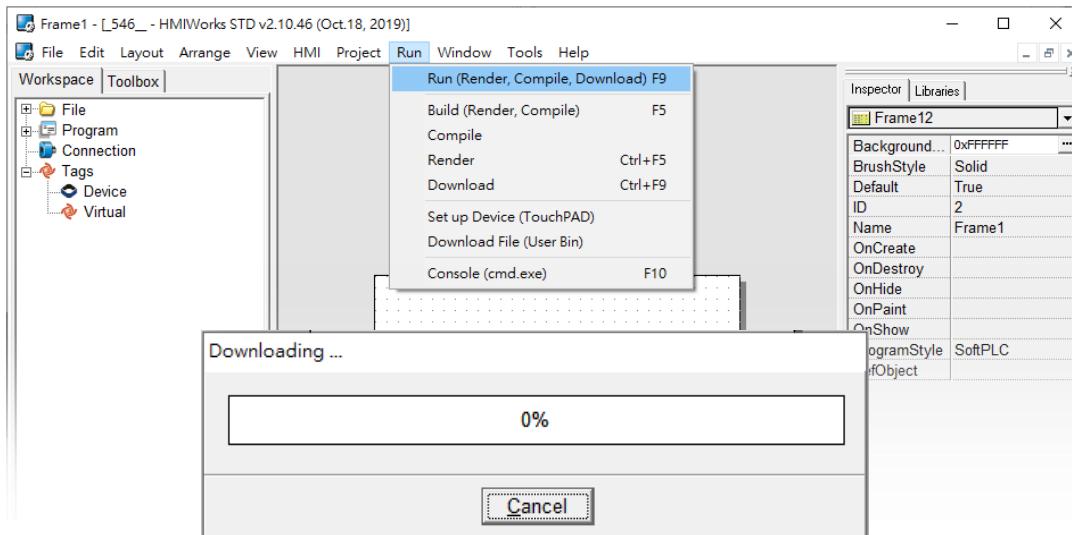


NOTE : If the TouchPAD can't be found, please power off the TouchPAD model first, then adjust the Rotary Switch to "1"(Ethernet Force Update mode), and try to search again.

Step 5: Return to the “Setup Ethernet Device” dialog box and enter an effective “IP address” in the “Download Information (TouchPAD)” field, and then click the “OK” button.



Step 6: In the HMIWorks application, click the “Run (Render, Compile, Download) F9” item from the “Run” menu, or press F9. The program will now be compiled.



Step 7: Wait until the Build and Download are completed(100%), the screen of the TouchPAD will display the project.



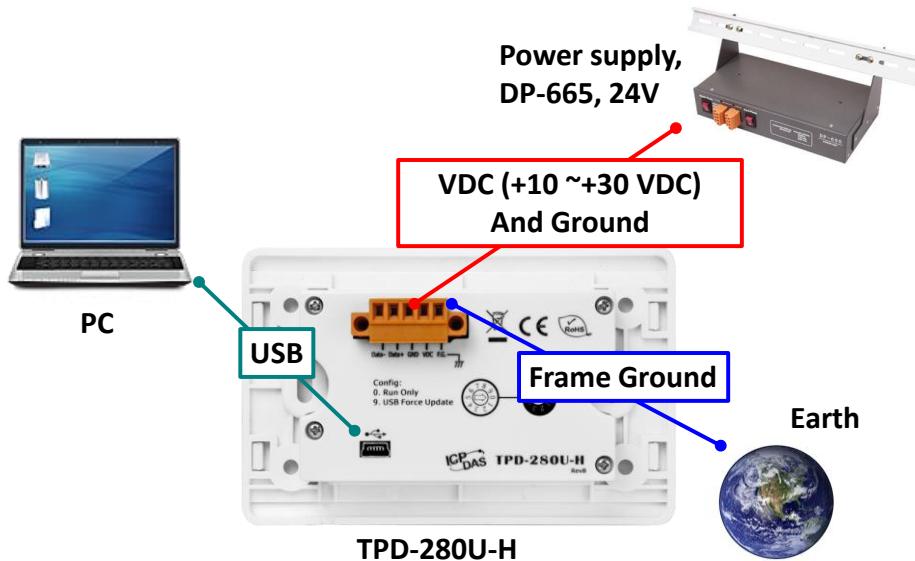
NOTE : If you use the “1”(Ethernet Force Update mode) to download, please adjust the Rotary Switch to “0”(Run mode) and reboot the TouchPAD model.

Applications are downloaded through **USB**.

Applicable Model:

- | | |
|---|--|
| <input checked="" type="checkbox"/> TPD-280U-H | <input checked="" type="checkbox"/> TPD-283U-H |
| <input checked="" type="checkbox"/> TPD-283U-M1 / M2 / M3 | <input checked="" type="checkbox"/> TPD-430-H(-EU) |
| <input checked="" type="checkbox"/> TPD-433-H(-EU) | <input checked="" type="checkbox"/> TPD-432F-H |
| <input checked="" type="checkbox"/> TPD-433F-M2 | <input checked="" type="checkbox"/> TPD-703(-64) |
| <input checked="" type="checkbox"/> VPD-130(N)-H | <input checked="" type="checkbox"/> VPD-132(N)-H |
| <input checked="" type="checkbox"/> VPD-133(N)-H | <input checked="" type="checkbox"/> VPD-142(N)-H |
| <input checked="" type="checkbox"/> VPD-143(N)-H | <input checked="" type="checkbox"/> VPD-173X(-64) |

Step 1: Connect the TouchPAD model to the Host PC using a CA-USB10 cable.

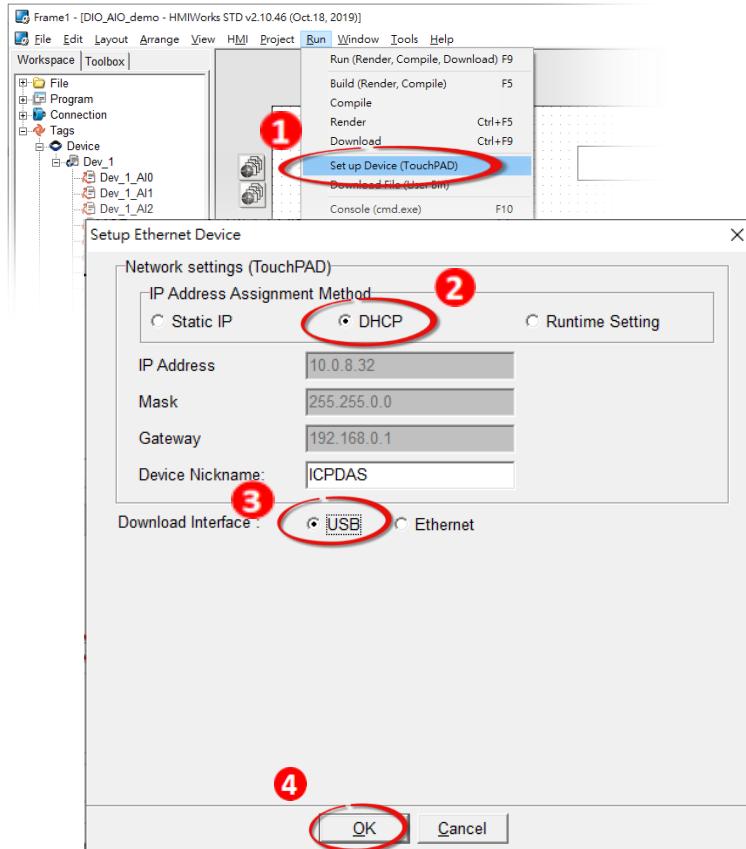


Step 2: Adjust the Rotary Switch to “9” (USB Force Update mode) and then reboot the TouchPAD model.



NOTE : The default setting of Rotary Switch on TouchPAD is “0” (Run mode)

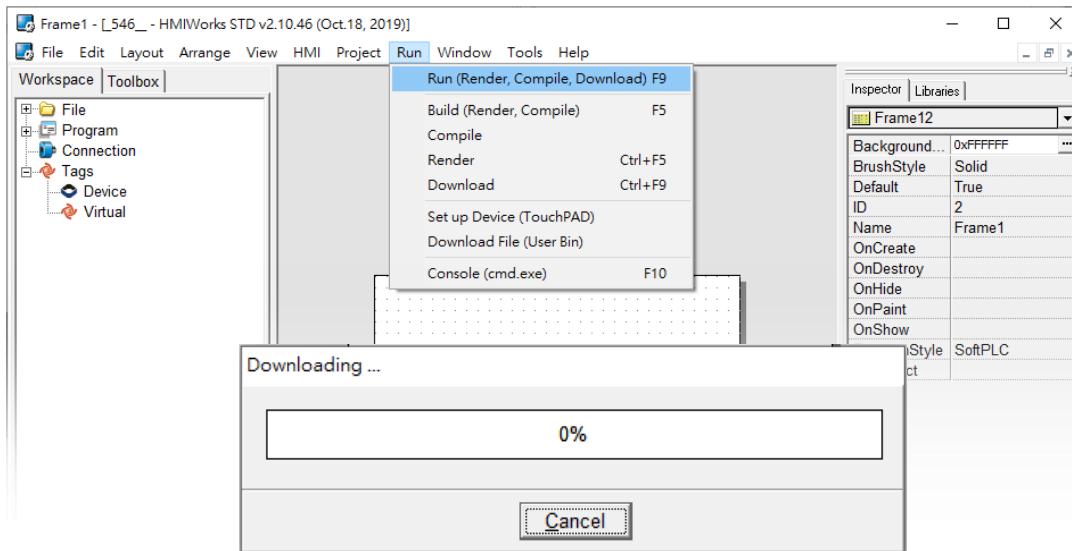
Step 3: Select the “Set up Device (TouchPAD)” item from the “Run” menu to open the “Setup Ethernet Device” dialog box. Select the “DHCP” and “USB” options, and then click the “OK” button to confirm setting.



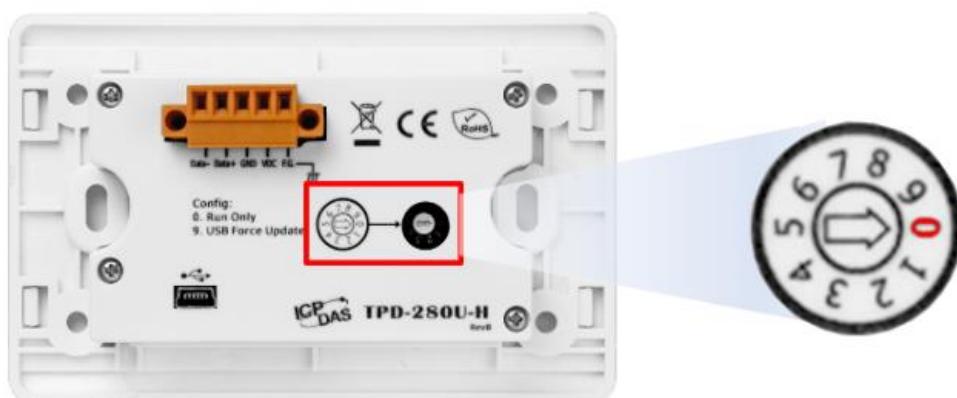
NOTE : If your network doesn't have access to a DHCP, you can use the Static IP and enter the effective IP information.
If the TouchPAD model didn't support the Ethernet function, you can skip this step.



Step 4: In the HMIWorks application, click the “Run (Render, Compile, Download) F9” item from the “Run” menu, or press F9. The program will now be compiled.



Step 5: Wait until the Build and Download are completed(100%). Adjust the Rotary Switch to “0” (Run mode), and then reboot the TouchPAD. After rebooting the TouchPAD, the screen of the TouchPAD will display the project.



Chapter 3. Demo Programs

In this chapter, you will be detailed instructions in how to write simple demo programs in either the C or LD language. Before beginning programming, please ensure that both the hardware and the software are installed ([Chapter 2](#)).

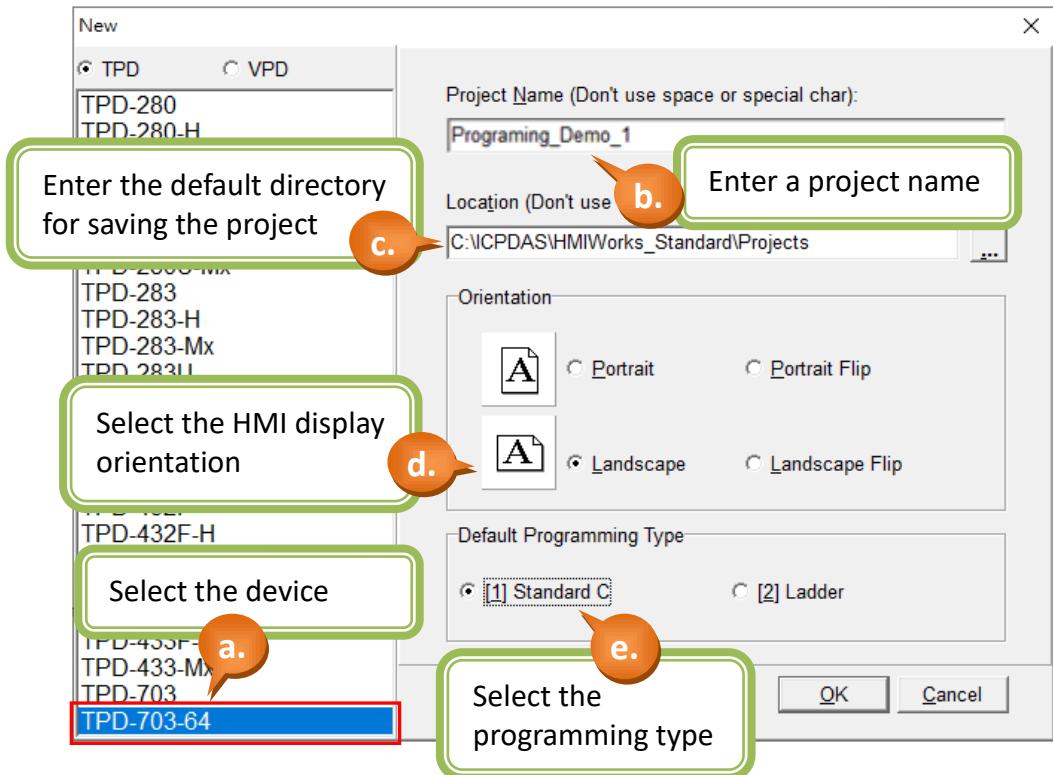
This chapter use the TPD-703-64 to design the demo.

3.1 Using the standard C language

3.1.1 Demo 1 (Beep, Timer, Counter)

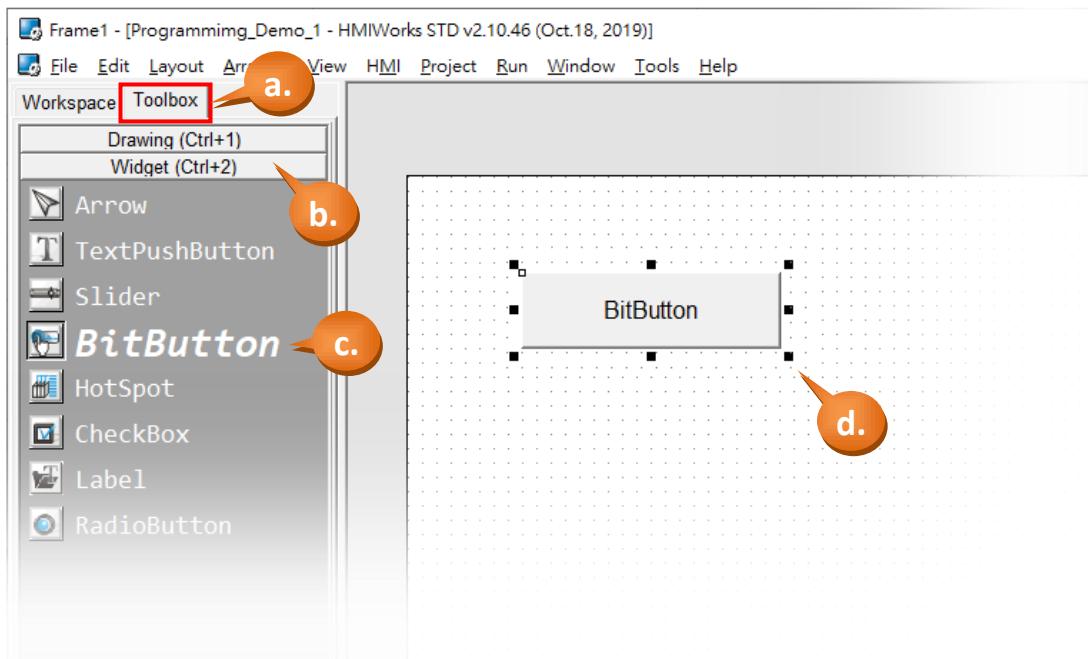
Demo 1 is programmed using the C language and implements a single button that will beep when clicked on and will increase the counter by one. Demo 1 also implements a second counter that will increase each second and then reset itself to zero when it reaches 100 seconds.

Step 1: Create a new TPD-703-64 project based on the C language (refer to [Section 2.2](#) for more details).

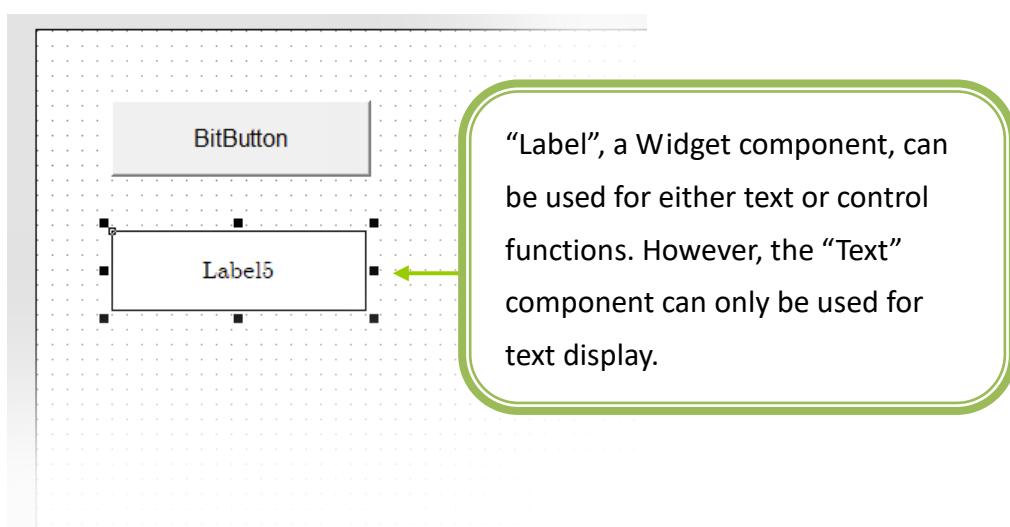


Step 2: Create a “BitButton” (picture button).

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “Widget (Ctrl+2)” heading to unfold the “Toolbox” panel.
- c. Select the “BitButton” icon, then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click and drag the mouse to draw a rectangle.

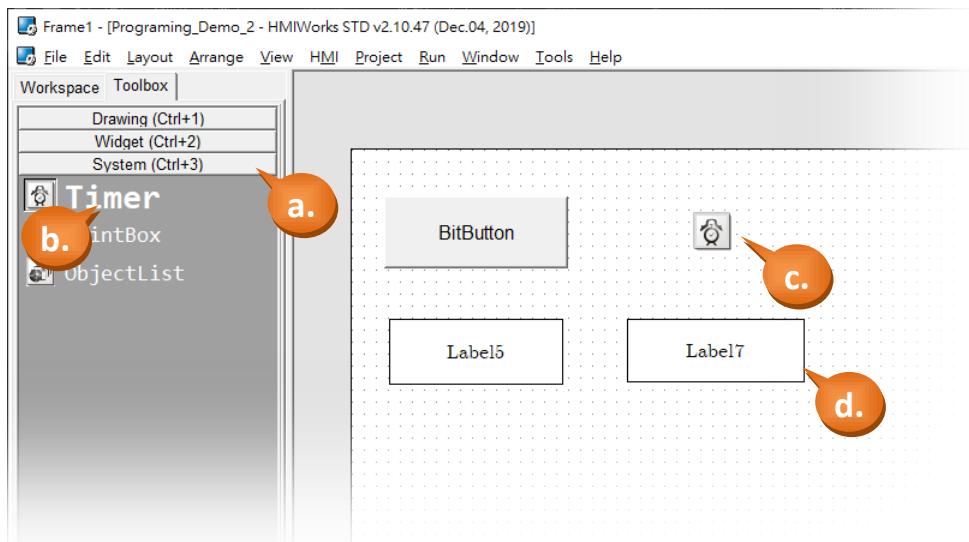


Step 3: In the same manner as the previous step, click on the “Label” icon and create a “Label” (text display) in the design area.



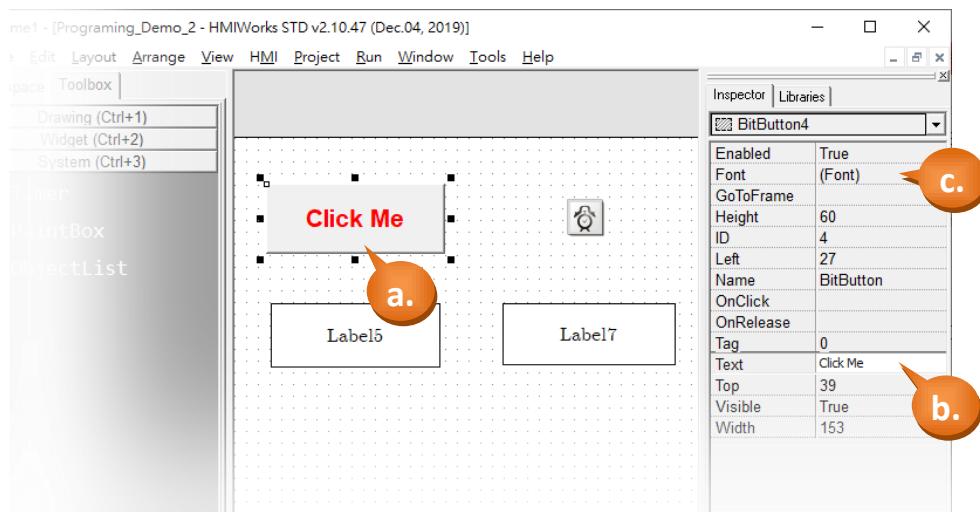
Step 4: Create a “Timer” and a “Label”.

- a. As shown in the picture below, click the “System (Ctrl+3)” heading to unfold the “Toolbox” panel.
- b. Select the “Timer” icon, then move your mouse to the Frame1 design area. You will notice that the cursor becomes a “+”.
- c. Click in the design area to create a Timer icon.
- d. In the same manner as that described in the Step 3, create another “Label”, as illustrated in the picture below.



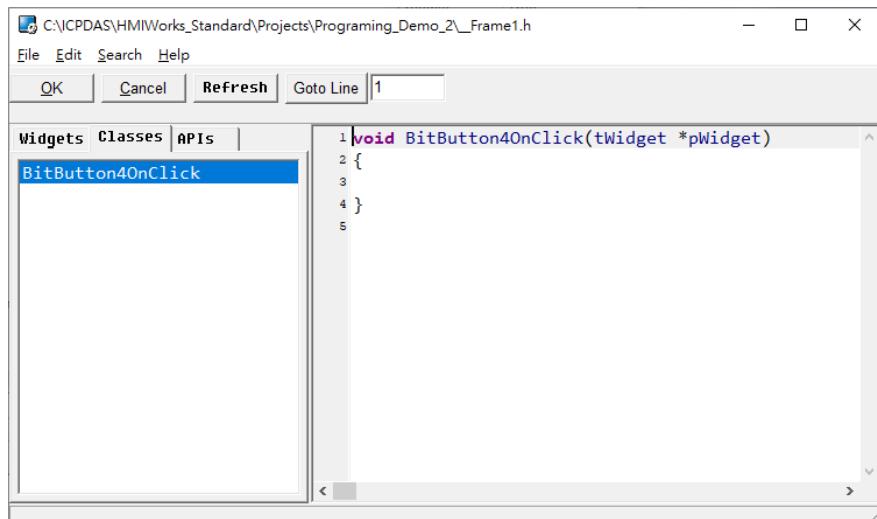
Step 5: Configure the properties for the “BitButton” component.

- a. Select the “BitButton” component in the design area.
- b. Click the “Text” field in the Properties Inspector panel and enter the string “Click Me”.
- c. Click the “Font” field to change the font to a suitable style, size and color.



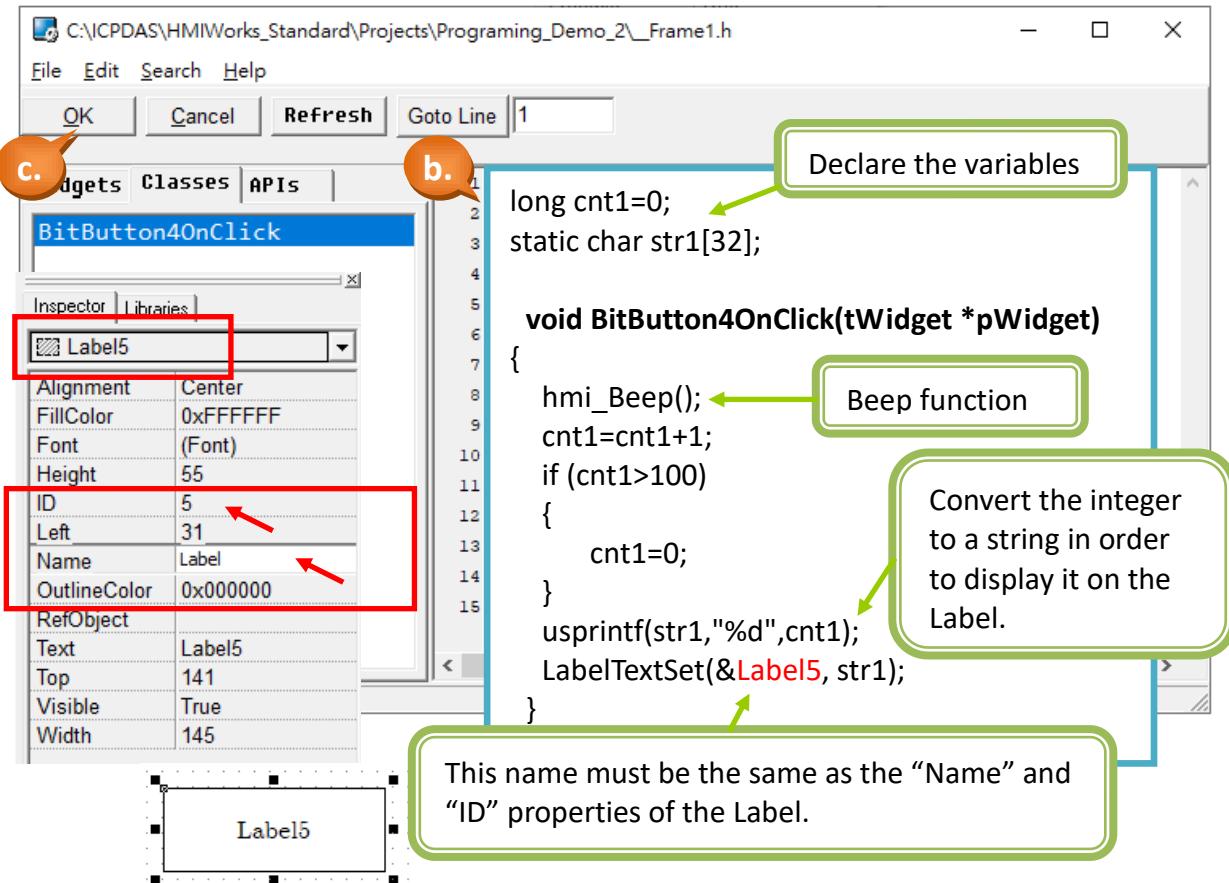
Step 6: Edit the control codes for the “BitButton” component.

- a. Double click the “BitButton” component in the design area to open the programming window.



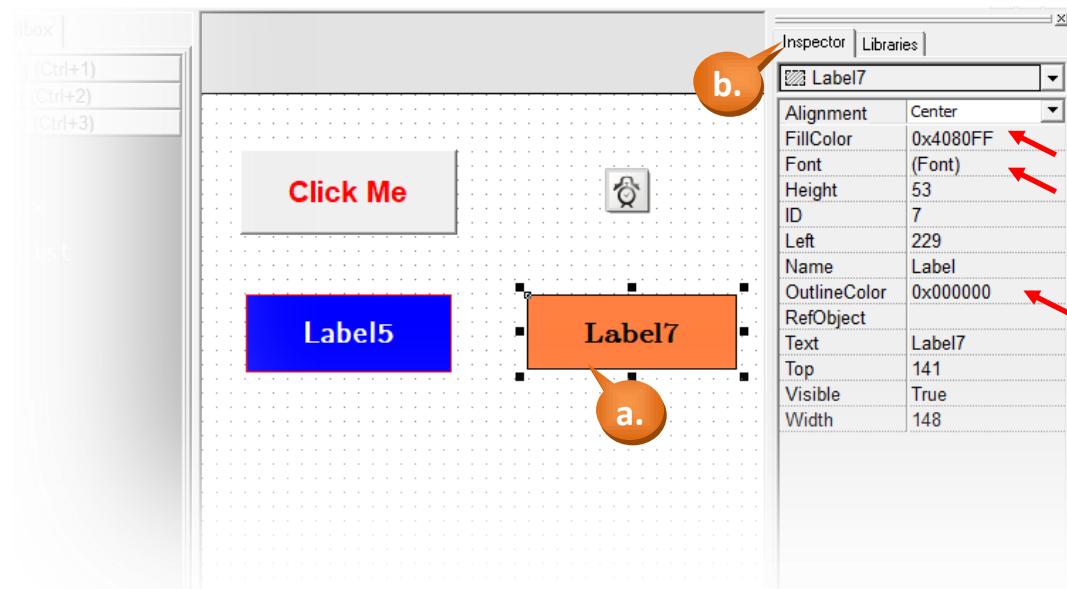
- b. Edit the code in the programming window using the C language. (Users can copy the code from the picture below.) In this demo, each click of the button will add 1 to the counter and beep once until it reaches 100, and will then reset to 0 before beginning to accumulate again.

- c. When finished, click “OK” to store.



Step 7: Configure the properties for the “Label” component.

- a. Select a Label ("Label5" or "Label7").



- b. The following properties can be modified from the Properites Inspector area on the right hand side:

Alignment: Sets the text alignment position in the display box of the “Label” component.

FillColor: Sets the fill color for the display box of the “Label” component.

Font: Sets the font for the text in the display box.

Height: Sets the height of the display box.

ID: A unique ID number to identify different entities of the same type in the design.

Left: Sets the X-coordinate of the top left corner of the display box.

Name: The type name of the component.

OutlineColor: Sets the outline color of the display box.

RefObject: Let the outward refer the picture in the ObjectList Widget.

Text: Sets the text to be displayed in the display box.

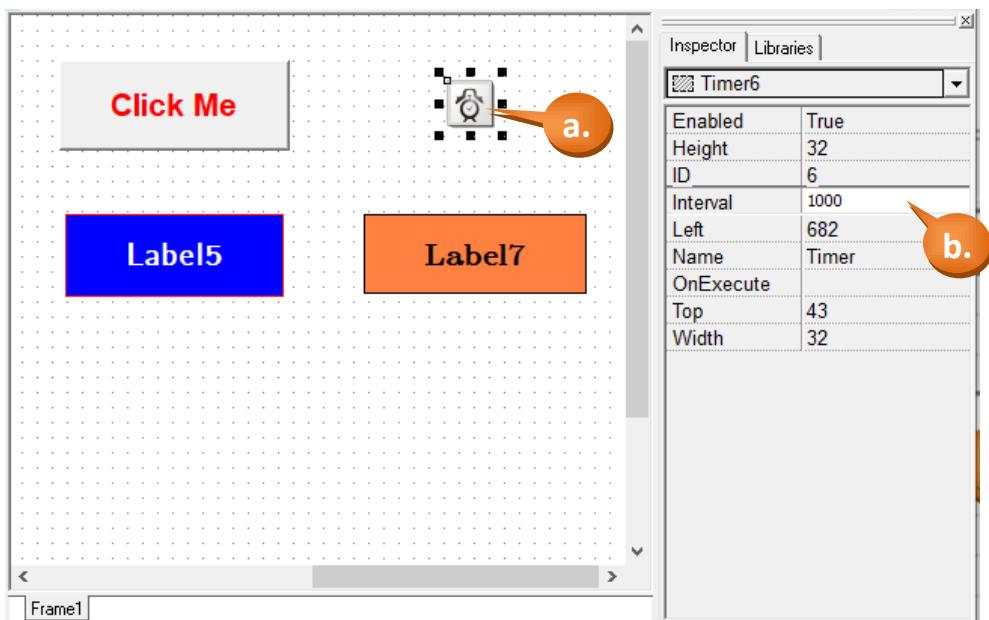
Top: Sets the Y-coordinate of the top left corner of the display box.

Visible: Sets the visibility of this Widget on the screen of model.

Width: Sets the width of the display box.

Step 8: Configure the properties of the “Timer” component.

- a. Select the “Timer6” component in the design area as shown in the picture below.
- b. Set the “Interval” property in the Properties Inspector area. The units are milliseconds (ms), so if the control function is to be executed every 1 second, set the value as “1000”.



Step 9: Edit the control codes for the “Timer” component.

- a. Double click the “Timer6” component in the design area to open the programming window.
- b. Edit the code in the programming window using the C language. (The User can copy the code from the picture below.) In this demo, the counter will increase by 1 each time the button is clicked until it reaches 100, and will then reset to 0 before beginning to accumulate again.
- c. When finished, click “OK” to store.

C:\ICPDAS\HMIWorks_Standard\Projects\Programming_Demo_2\Frame1.h

File Edit Search Help

OK Cancel Refresh Got 1

Widgets Classes APIs

Timer6OnExecute BitButton40OnClick

Label7

Alignment Center

FillColor 0x4080FF

Font (Font)

Height 53

ID 7

Left 627

Name Label

OutlineColor 0x000000

RefObject

Text Label7

Top 132

Visible True

Width 148

long cnt1=0;
long auto_cnt1=0;
static char str1[32];

void Timer6OnExecute(tWidget *pWidget)
{
 auto_cnt1=auto_cnt1+1;
 if (auto_cnt1>100)
 {
 auto_cnt1=0;
 }
 usprintf(str1,"%d",auto_cnt1);
 LabelTextSet(&Label7,str1);
}

b.

Declare the variable (auto_cnt1)

Convert the integer to a string in order to display it on the Label.

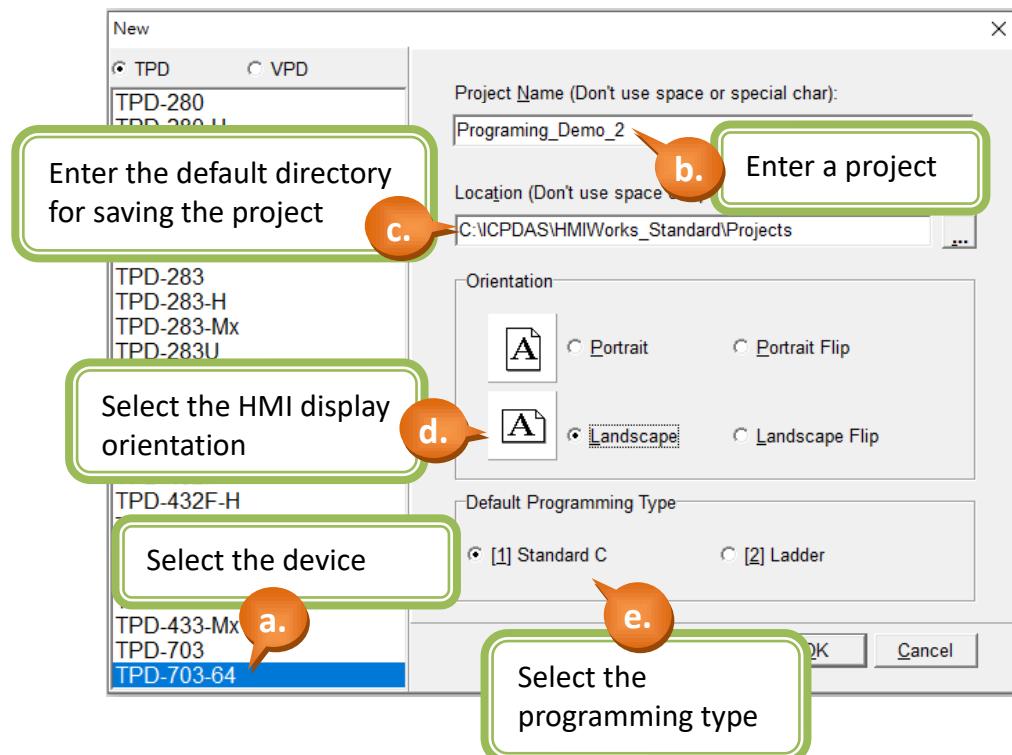
This name must be the same as “Name” and “ID” properties of the Label.

Step 10: Build and Download the demo to the TouchPAD model.
Please refer [2.4 Downloading Methods for TouchPAD](#)

3.1.2 Demo 2 (Use Slider to control the object)

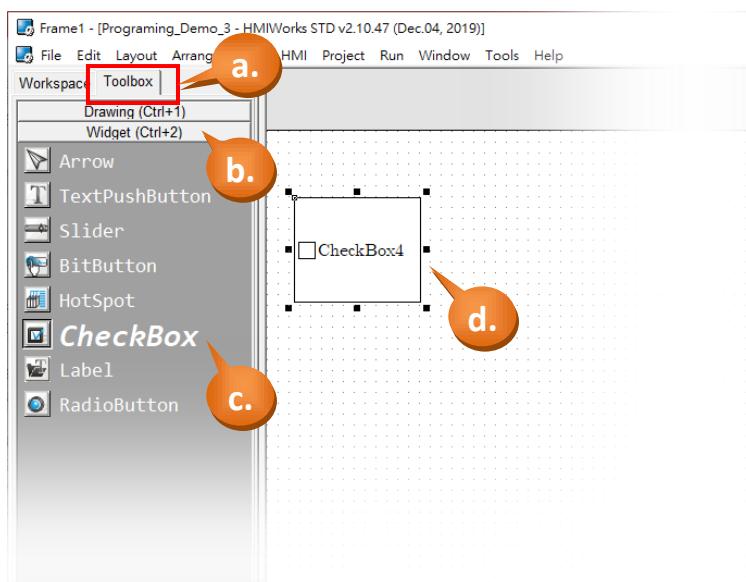
Demo 2 is programmed using the C language. This demo combine the CheckBox and ObjectList to implement the effect of image changes, and the CheckBox will be change by the value of the Slider.

Step 1: Create a new TPD-703-64 project based on the C language (refer to [Section 2.2](#) for more details).



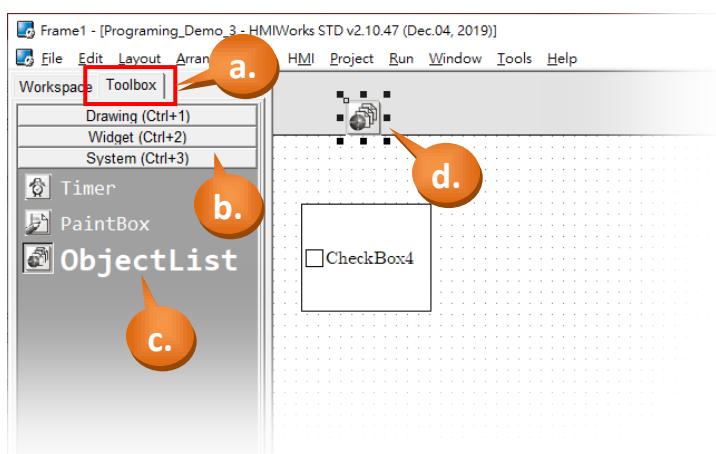
Step 2: Create a “CheckBox” (Choice button).

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “Widget (Ctrl+2)” heading to unfold the “Toolbox” panel.
- c. Select the “CheckBox” icon, and then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click and drag the mouse to draw a rectangle.



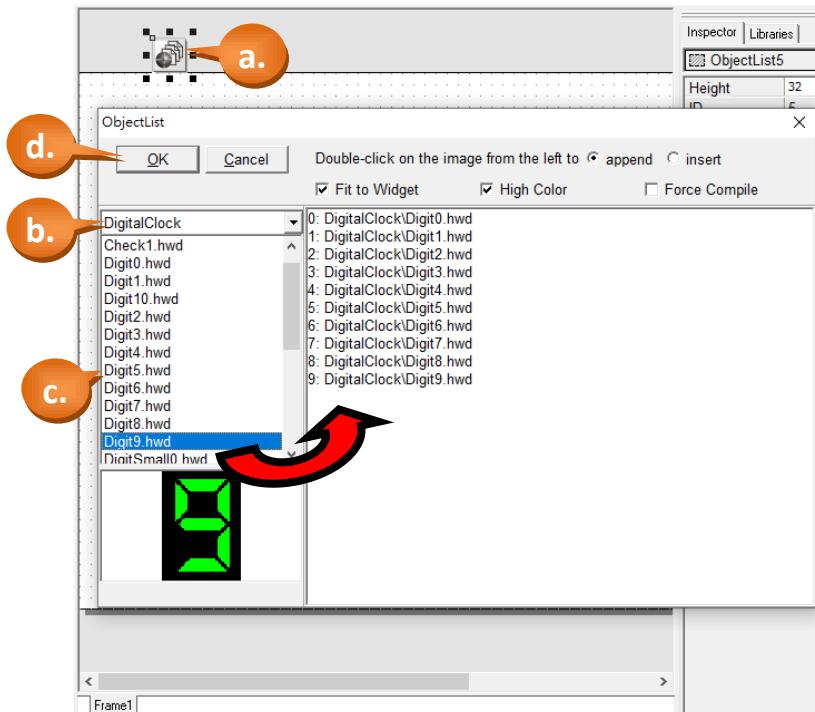
Step 3: Create an “ObjectList” (Object List).

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “System (Ctrl+3)” heading to unfold the “Toolbox” panel.
- c. Select the “ObjectList” icon, and then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click the mouse to place an “ObjectList”.



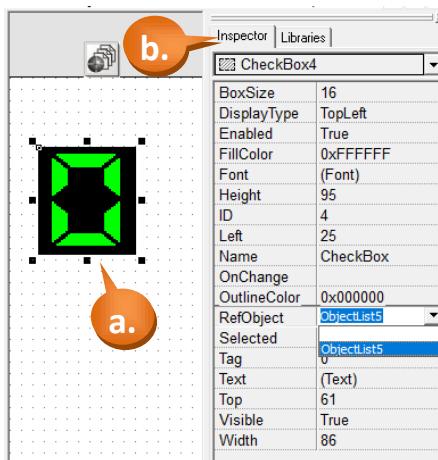
Step 4: Create the list of picture in the “ObjectList”.

- a. Double click the “ObjectList5” to open the setting dialog box.
- b. Select the “DigitalClock” in the drop-down menu.
- c. Double click the Digit0.hwd to Digit9.hwd in order; the list will be created on right block.
- d. Click “OK” button to complete the list of pictures.



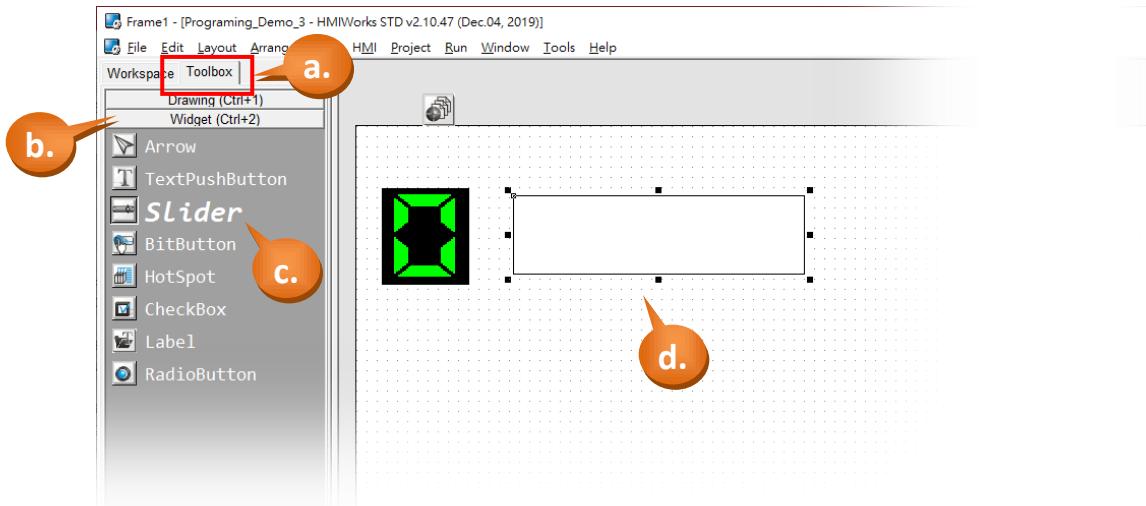
Step 5: Connect the “ObjectList” and the “CheckBox” widget.

- a. Click the “CheckBox4”, as shown in the picture below.
- b. Select the “RefObject” property in the Property Inspector, and select the “ObjectList5” in the drop-down menu, then the “CheckBox4” will display digit 0, as shown in the picture below.



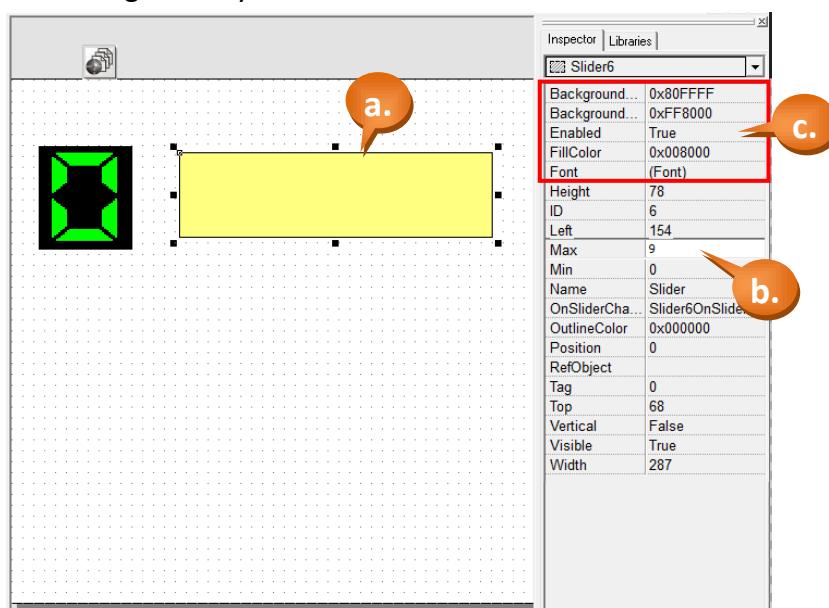
Step 6: Create a “Slider”.

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “Widget (Ctrl+2)” heading to unfold the “Toolbox” panel.
- c. Select the “Slider” icon, and then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click and drag the mouse to draw a rectangle.



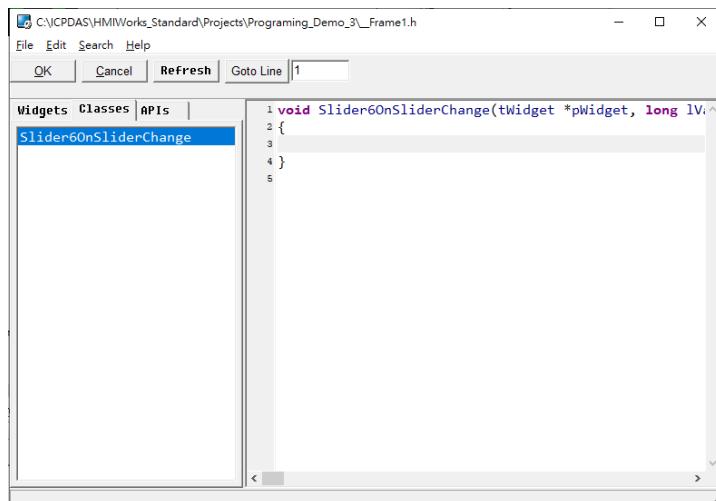
Step 7: Adjust the range and the style of the Slider.

- a. Click the “Slider6”, as shown in the picture below.
- b. Modify the value of “Max” to “9” in the Property Inspector.
- c. Click the “BackgroundFillColor”, the “BackgroundTextColor”, the “FillColor” and the “Font” field to change the style.



Step 8: Edit the control codes for the “Slider” component.

- a. Double click the “Slider” component in the design area to open the programming window.



- b. Edit the code in the programming window using the C language. (Users can copy the code from the picture below.) In this demo, the “CheckBox4” will be changed by the value of “Slider6”. Both the text in the “Slider6” and the corresponding picture will be changed at the same time.

- c. When finished, click “OK” to store.

The screenshot shows the same programming window as above, but with annotations:

- C.** Points to the "Widgets" tab in the left sidebar.
- b.** Points to the code area where the function "Slider6OnSliderChange" is defined.
- Declare the variables**: A callout points to the declaration of the character array "strValue".
- Convert the integer to a string in order to display it on the Slider.**: A callout points to the line "usnprintf(strValue, sizeof(strValue), "%d", lValue);".
- This name must be the same as the “Name” and “ID” properties of the CheckBox.**: A callout points to the line "SliderTextSet(&Slider6, strValue);".

The left sidebar shows the properties of a "CheckBox4" component, with the "Name" and "ID" properties highlighted.

```

void Slider6OnSliderChange(tWidget *pWidget, long lValue)
{
    static char strValue[10];
    usnprintf(strValue, sizeof(strValue), "%d", lValue);
    SliderTextSet(&Slider6, strValue);
    WidgetPaint((tWidget*)&Slider6);

    CheckBoxValueSet(&CheckBox4, lValue);
}

```

Step 9: Build and Download the demo to the TouchPAD model.

Please refer [2.4 Downloading Methods for TouchPAD](#)

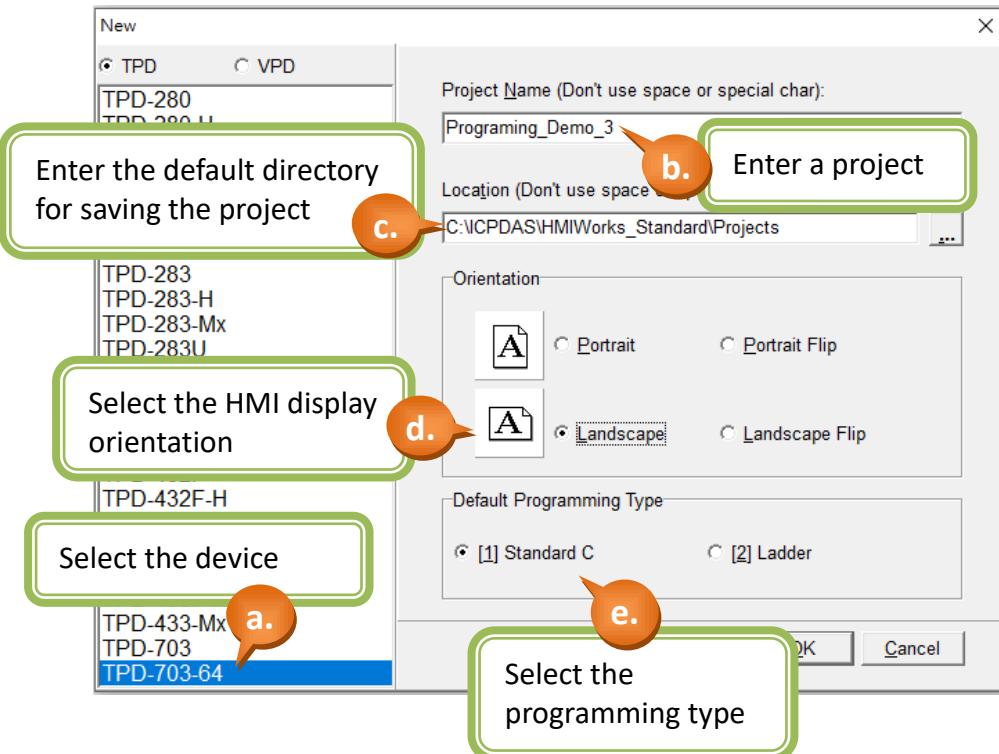
3.1.3 Demo 3 (Digital Clock)

Demo 3 is programmed using the C language and implements a digital clock that will update the real time using built-in RTC. And this demo shows how to add the background which is designed by user.



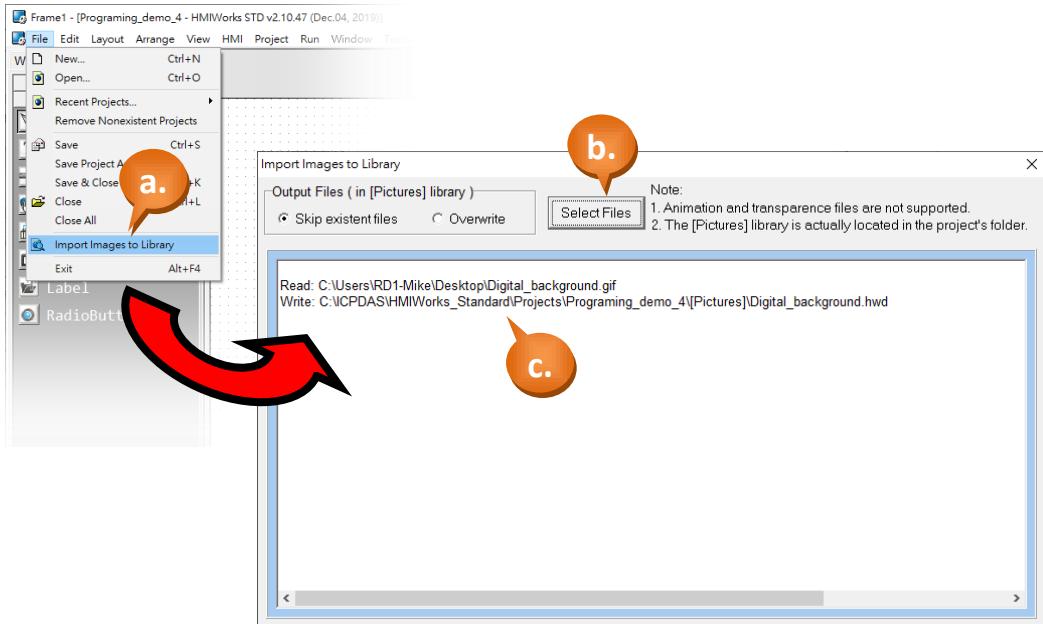
NOTE : Demo 3 use the built-in RTC in the TPD/VPD model, the TPD-280-H/Mx and TPD-283-H/Mx model are not support the RTC function.

Step 1: Create a new TPD-703-64 project based on the C language (refer to [Section 2.2](#) for more details).



Step 2: Import a picture as the background.

- a. Select the “Import Images to Library” in the “File” menu to open the dialog box.
- b. Click the “Select Files” button and choice the picture, i.e., the background.
- c. After choosing the picture, the dialog box will show the path of related file, and that means the importation is completed.

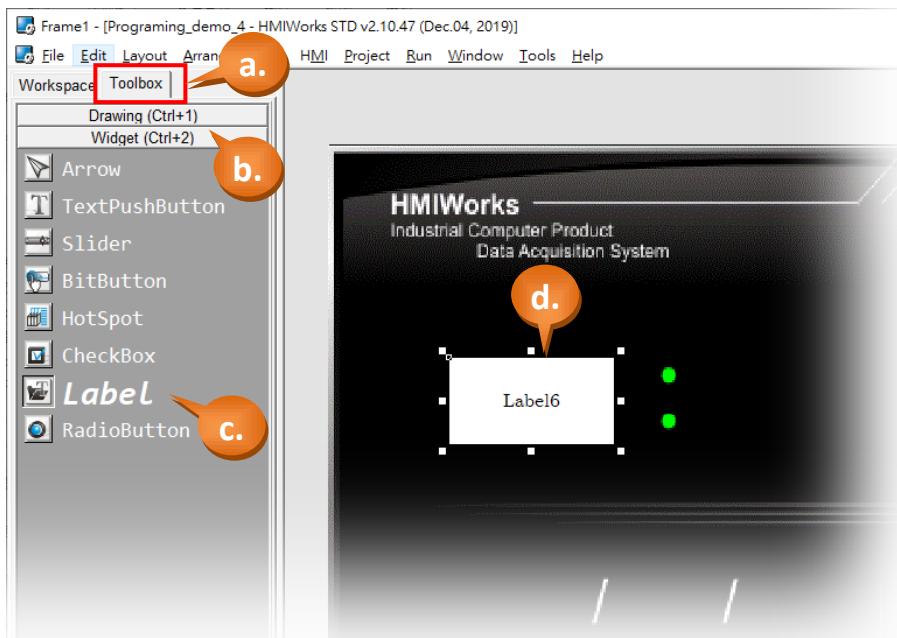


- d. Close the “Import Images to Library” dialog box and click the “Libraries”
- e. Select the “Pictures” category and you will find the imported picture.
- f. Select and drag the imported picture to the design area.

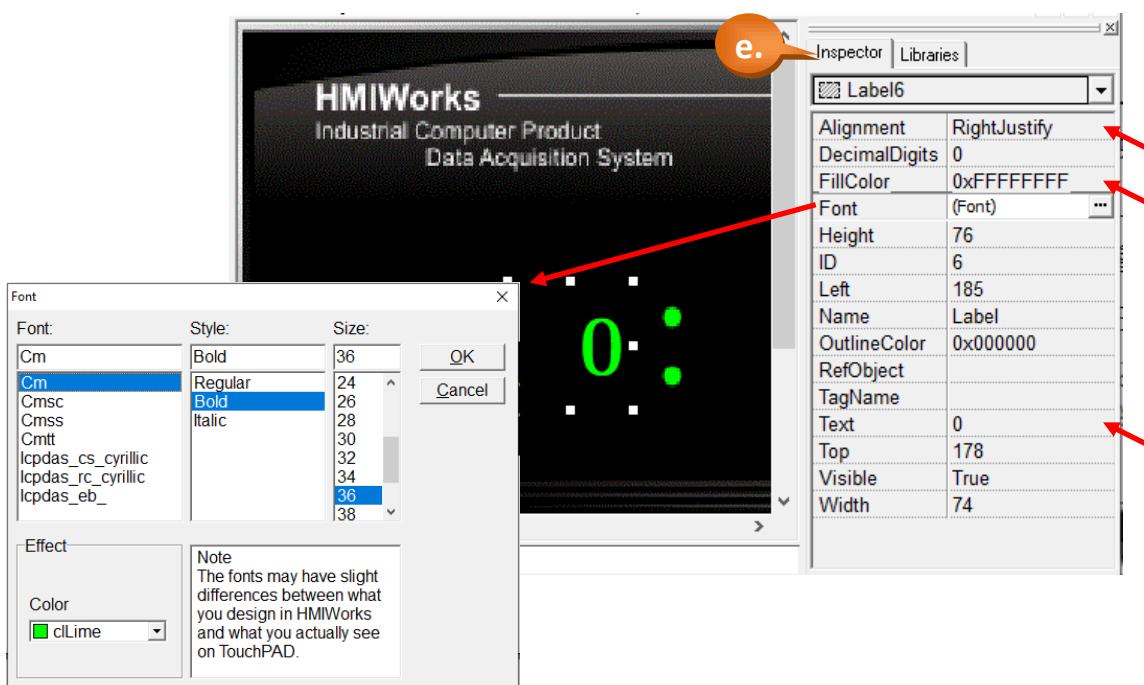


Step 3: Create a “Label” (text display).

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “Widget (Ctrl+2)” heading to unfold the “Toolbox” panel.
- c. Select the “Label” icon, then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click and drag the mouse to draw a rectangle.



- e. Click the “Alignment”, the “FillColor”, the “Font” and the “Text” field to change the style from the Property Inspector, as shown in the picture below.

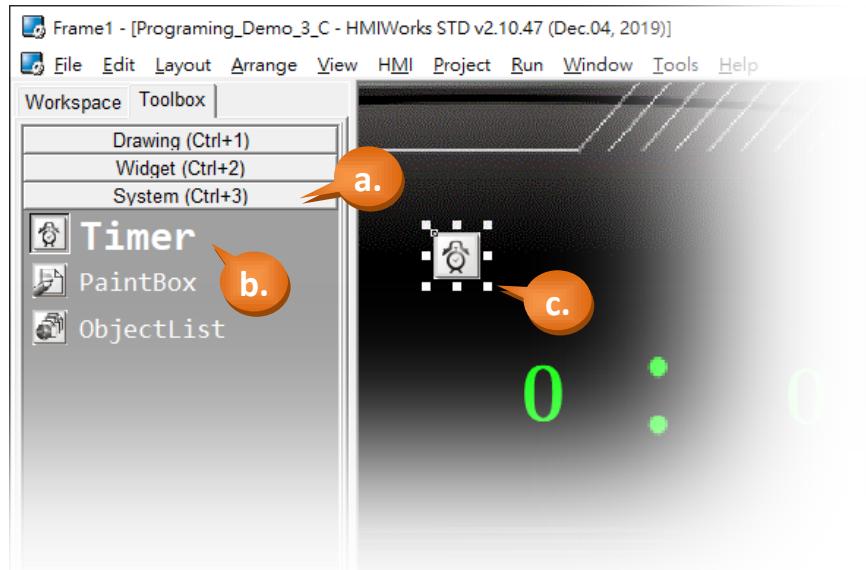


Step 4: Repeat the step 3 to create others Label.



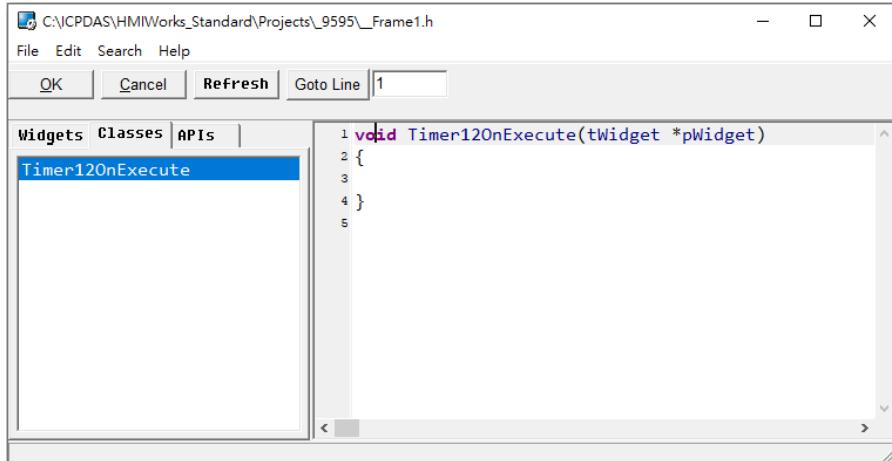
Step 5: Create a “Timer”.

- As shown in the picture below, click the “System (Ctrl+3)” heading to unfold the “Toolbox” panel.
- Select the “Timer” icon, then move your mouse to the Frame1 design area. You will notice that the cursor becomes a “+”.
- Click in the design area to create a Timer icon.



Step 6: Edit the control codes for the “Timer” component.

- a. Double click the “Timer” component in the design area to open the programming window.



- b. Edit the code in the programming window using the C language. (Users can copy the code from the picture below.) In this demo, we use the “hmi_GetDateTime” API to obtain the value of the RTC and display them in corresponding Label.

- c. When finished, click “OK” to store.

The screenshot shows the same programming window with annotations:

- A callout labeled "c." points to the "Widgets" tab in the left pane.
- A callout labeled "b." points to the code area. A green box labeled "Declare the variables" encloses the declarations:

```
int yy,mm,dd,hh,nn,ss;
static char str1[5],str2[5],str3[5],str4[5],str5[5],str6[5];
```
- An arrow points from the "Get the value of the RTC" callout to the line: `hmi_GetDateTime(&yy,&mm,&dd,&hh,&nn,&ss);`
- An arrow points from the "Convert the integer to a string in order to display it on the Label" callout to the line: `usprintf(str1,"%d",yy);`

```
int yy,mm,dd,hh,nn,ss;
static char str1[5],str2[5],str3[5],str4[5],str5[5],str6[5];

void Timer12OnExecute(tWidget *pWidget)
{
    hmi_GetDateTime(&yy,&mm,&dd,&hh,&nn,&ss);

    usprintf(str1,"%d",yy);
    LabelTextSet(&Label11,str1);

    usprintf(str2,"%d",mm);
    LabelTextSet(&Label10,str2);

    usprintf(str3,"%d",dd);
    LabelTextSet(&Label9,str3);

    usprintf(str4,"%d",hh);
    LabelTextSet(&Label8,str4);

    usprintf(str5,"%d",nn);
    LabelTextSet(&Label7,str5);

    usprintf(str6,"%d",ss);
    LabelTextSet(&Label6,str6);
}
```

Step 7: Build and Download the demo to the TouchPAD model.
Please refer [2.4 Downloading Methods for TouchPAD](#)



3.1.4 Demo 4 (Remotely control the ICPDAS I/O model)

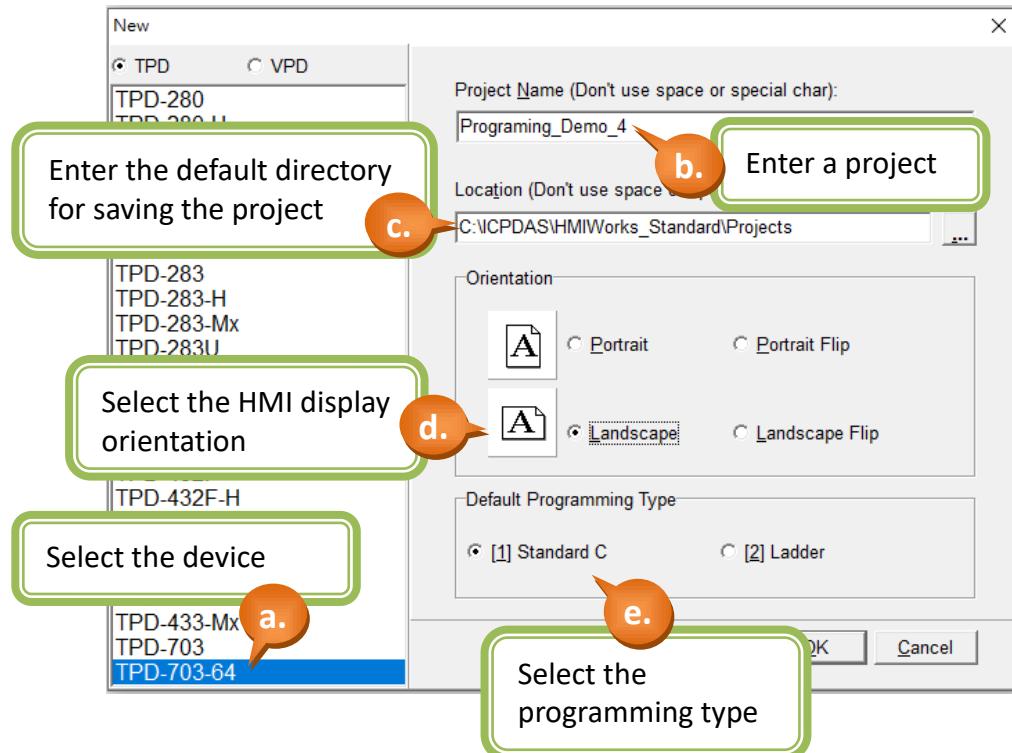
Demo 4 is programmed using the C language and implements that remotely control the ICPDAS I/O model.

In this demo, we use the ICPDAS M-7055 model to demonstrate. This demo will use the switch on the screen of the TPD model to control the DO of the M-7055 model, and the TPD also update the DI status of the M-7055 model at any time on its screen.

Please connect the DI0 pin and the DO0 pin of the M-7055 model. Please refer the manual of M-7055 to view the detailed wiring.

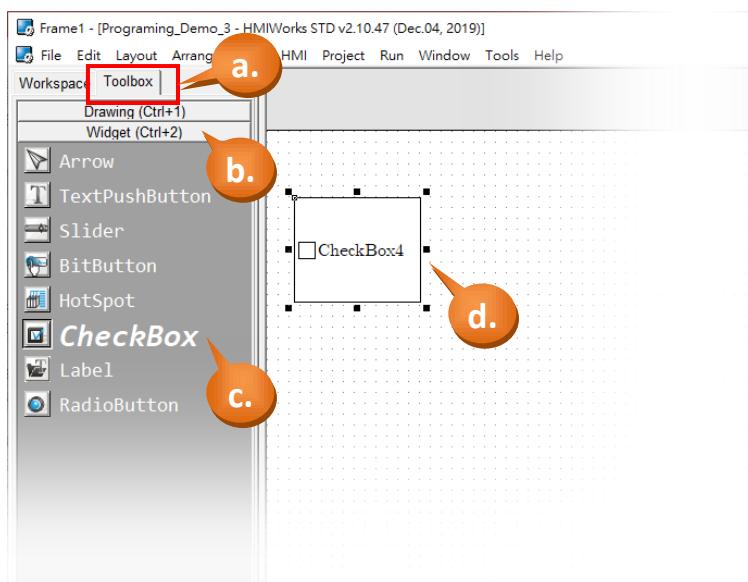
http://www.icpdas.com/web/product/download/io_and_unit/rs-485/document/manual/7000/I-7000_M-7000_DIO_en.pdf

Step 1: Create a new TPD-703-64 project based on the C language (refer to [Section 2.2](#) for more details).



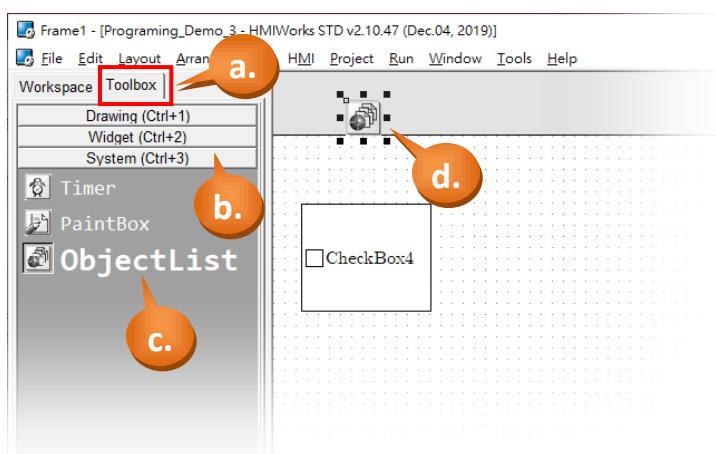
Step 2: Create a “CheckBox” (Choice button) to display the status of DI0.

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “Widget (Ctrl+2)” heading to unfold the “Toolbox” panel.
- c. Select the “CheckBox” icon, and then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click and drag the mouse to draw a rectangle.



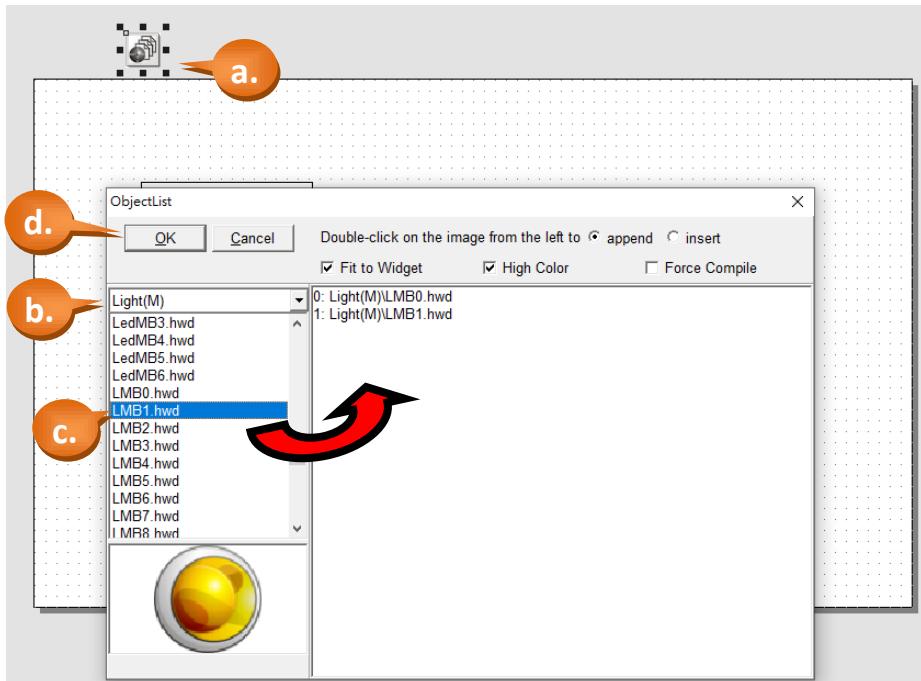
Step 3: Create an “ObjectList” (Object List).

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “System (Ctrl+3)” heading to unfold the “Toolbox” panel.
- c. Select the “ObjectList” icon, and then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click the mouse to place an “ObjectList”.



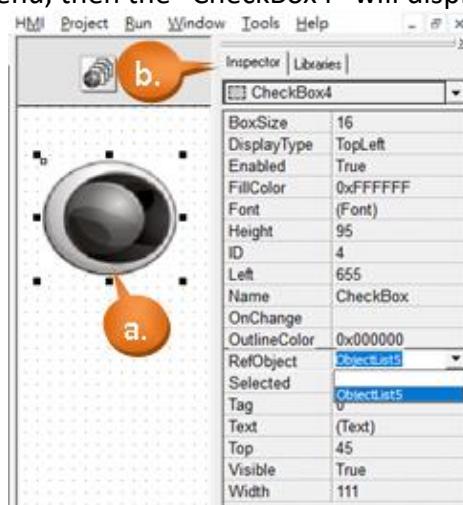
Step 4: Create the list of picture in the “ObjectList”.

- a. Double click the “ObjectList5” to open the setting dialog box.
- b. Select the “Light(M)” in the drop-down menu.
- c. Double click the LMB0.hwd and then double click the LMB1.hwd, the list will be created on right block.
- d. Click “OK” button to complete the list of pictures.

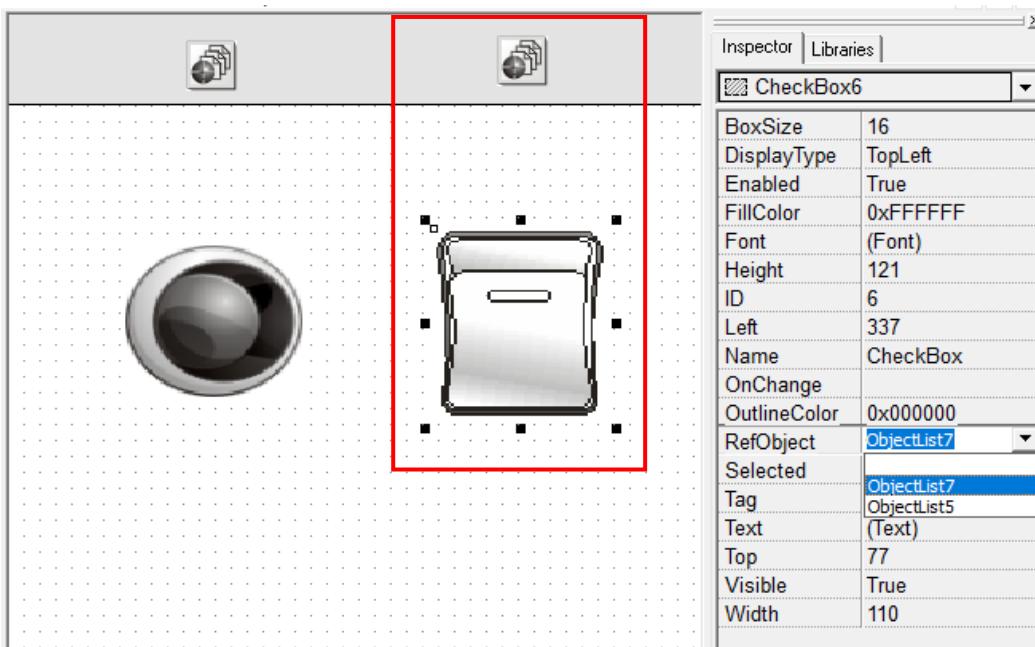


Step 5: Connect the “ObjectList” and the “CheckBox” widget.

- a. Click the “CheckBox4”, as shown in the picture below.
- b. Select the “RefObject” property in the Property Inspector, and select the “ObjectList5” in the drop-down menu, then the “CheckBox4” will display the light.

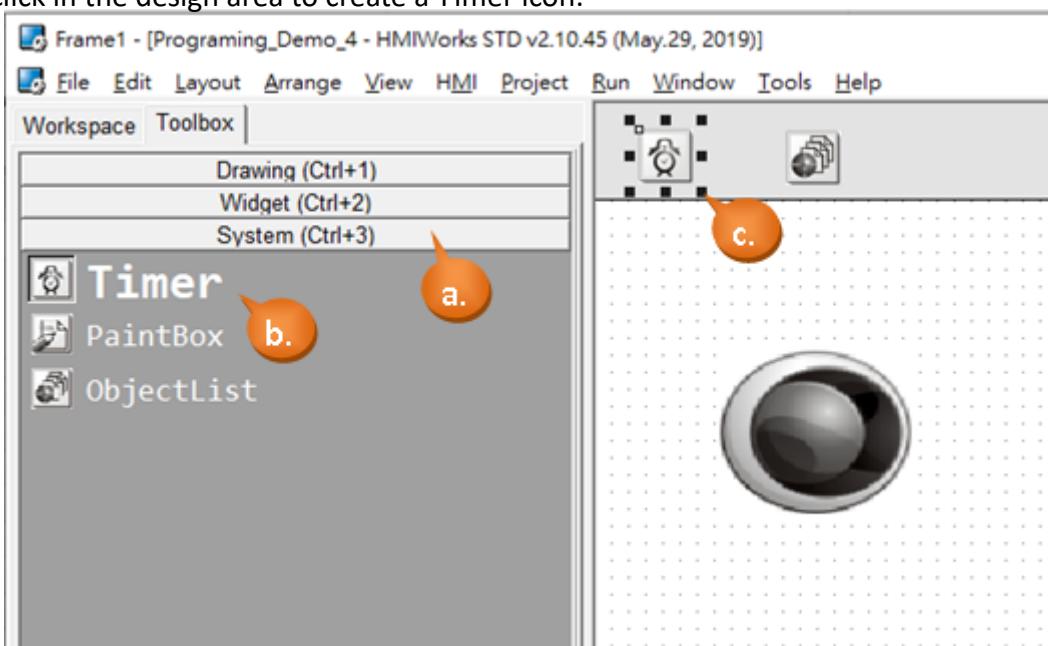


Step 6: Repeat Step 2 to Step 5 to create a button which designed to control the DO0.



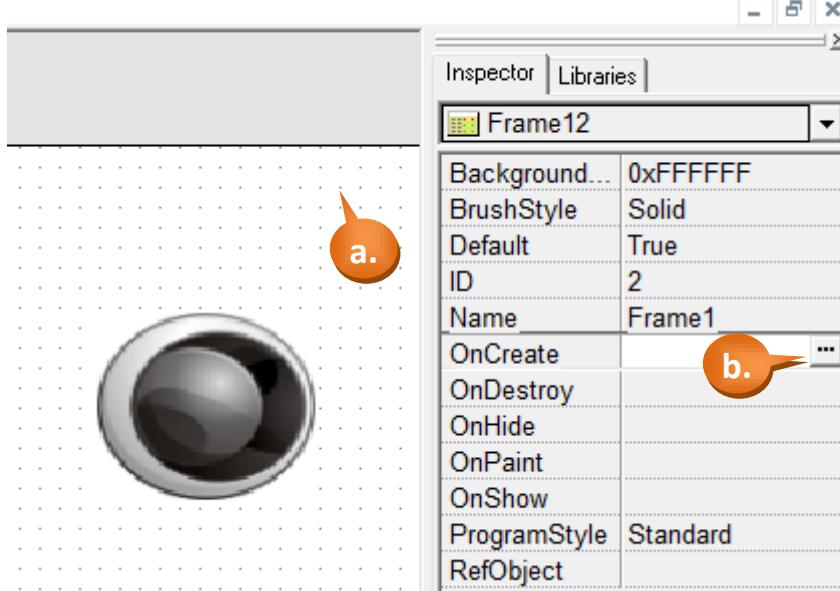
Step 7: Create a “Timer”.

- As shown in the picture below, click the “System (Ctrl+3)” heading to unfold the “Toolbox” panel.
- Select the “Timer” icon, then move your mouse to the Frame1 design area. You will notice that the cursor becomes a “+”.
- Click in the design area to create a Timer icon.



Step 8: Edit the control codes for the “Frame1” component.

- Click the blank at the design area to open the Property Inspector of the “Frame12”.
- Click the “...” button at “OnCreate” field to open the programming window.

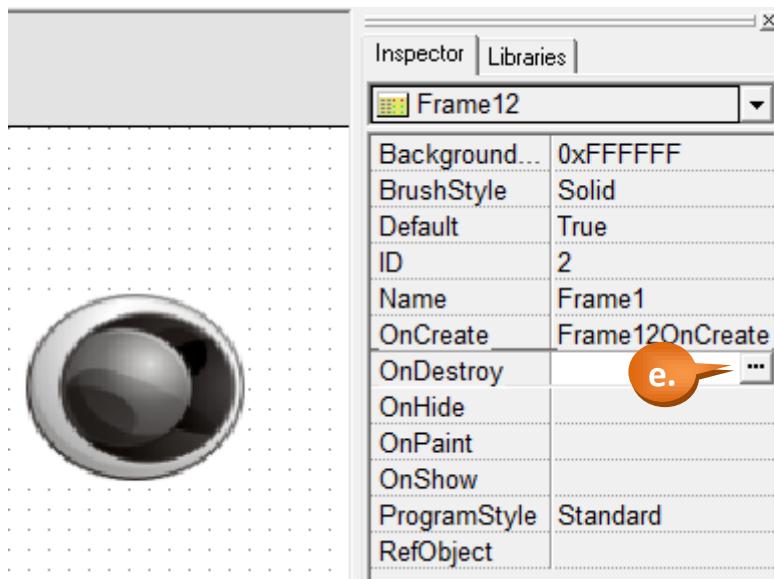


- Edit the code in the programming window using the C language. (Users can copy the code from the picture below.)
- When finished, click “OK” to store.

```
File Edit Search Help
OK Cancel Refresh Goto Line 1
Widgets Classes APIs
Frame12OnCreate
HANDLE h;

void Frame12OnCreate()
{
    h = uart_Open("COM1,9600,N,8,1");
}
```

- e. Click the “...” button at “OnDestroy” field to open the programming window.



- f. Edit the code in the programming window using the C language. (Users can copy the code from the picture below.)
g. When finished, click “OK” to store.

```

C:\ICPDAS\HMIWorks_Standard45\Projects\Programming_Demo_4\Frame1.h
File Edit Search Help
OK Cancel Refresh Goto Line 1
Widgets Classes APIs | f.
Frame12OnCreate
Frame12OnDestroy
HANDLE h;

void Frame12OnCreate()
{
    h = uart_Open("COM1,9600,N,8,1");
}
void Frame12OnDestroy()
{
    uart_Close(h); g.
}

```

When the Frame12 is closed, closing the COM Port.

Step 9: Edit the control codes for the “Timer” component.

- Double click the “Timer” component in the design area to open the programming window.

The screenshot shows a programming window with the file path C:\ICPDAS\HMIWorks_Standard45\Projects\Programming_Demo_4\Frame1.h. The code editor displays the following C language code:

```

1 HANDLE h;
2
3 void Frame12OnCreate()
4 {
5     h = uart_Open("COM1,9600,N,8,1");
6 }
7
8 void Frame12OnDestroy()
9 {
10    uart_Close(h);
11 }
12
13
14 void Timer80OnExecute(tWidget *pWidget)
15 {
16
17 }
18

```

The 'Widgets' tab is selected in the left sidebar, and the 'Timer80OnExecute' function is highlighted in the code.

- Edit the code in the programming window using the C language. (Users can copy the code from the picture below.)
- When finished, click “OK” to store.

The screenshot shows the same programming window with the following C language code. Annotations explain specific parts of the code:

```

HANDLE h;
char ReadDI_value[8];
int NetID = 1;
int addr = 0;
int ch_count = 8;

void Frame12OnCreate()
{
    h = uart_Open("COM1,9600,N,8,1");
}

void Frame12OnDestroy()
{
    uart_Close(h);
}

void Timer10OnExecute(tWidget *pWidget)
{
    mrm_ReadDI (h, NetID, addr, ch_count, ReadDI_value);

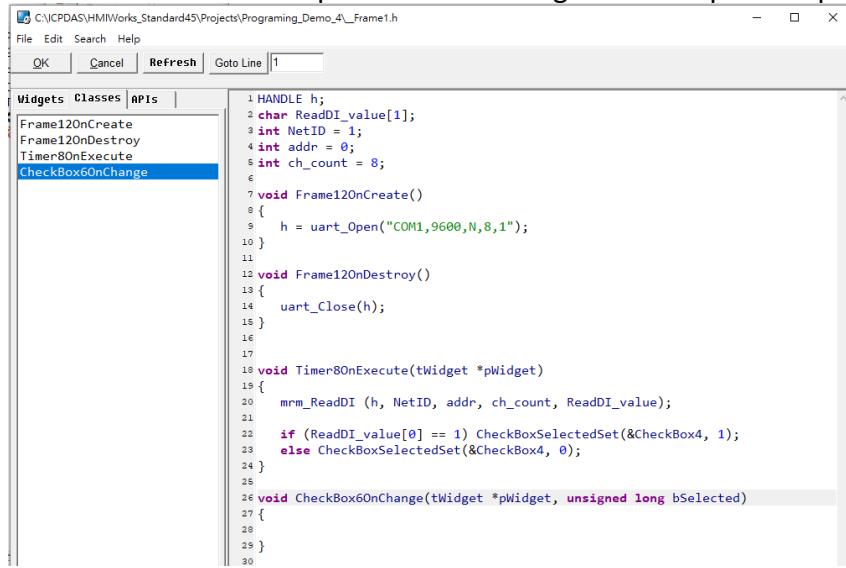
    if (ReadDI_value[0] == 1) CheckBoxSelectedSet(&CheckBox4, 1);
    else CheckBoxSelectedSet(&CheckBox4, 0);
}

```

- C.** Points to the 'Timer10OnExecute' function in the code editor.
- b.** Points to the declaration of variables: `HANDLE h;`, `char ReadDI_value[8];`, `int NetID = 1;`, `int addr = 0;`, and `int ch_count = 8;`. A callout box states: "Declare the related variables of M-7055 model."
- A callout box points to the `mrm_ReadDI` call in the `Timer10OnExecute` function: "Read the DI status of the model regularly and change the light according to the reading."

Step 10: Edit the control codes for the Button component.

- a. Double click the “CheckBox6” component in the design area to open the programming window.



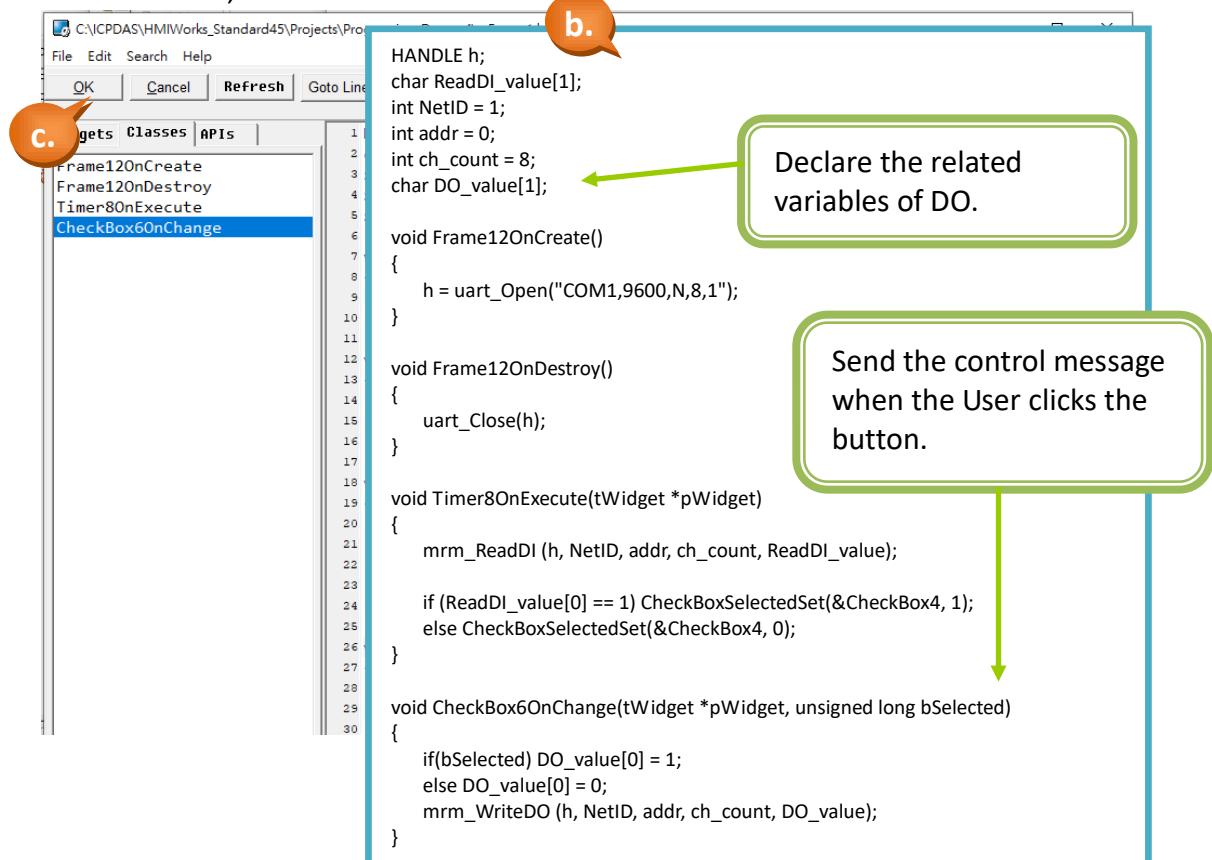
```

1 HANDLE h;
2 char ReadDI_value[1];
3 int NetID = 1;
4 int addr = 0;
5 int ch_count = 8;
6
7 void Frame12OnCreate()
8 {
9     h = uart_Open("COM1,9600,N,8,1");
10 }
11
12 void Frame12OnDestroy()
13 {
14     uart_Close(h);
15 }
16
17
18 void Timer8OnExecute(tWidget *pWidget)
19 {
20     mrm_ReadDI (h, NetID, addr, ch_count, ReadDI_value);
21
22     if (ReadDI_value[0] == 1) CheckBoxSelectedSet(&CheckBox4, 1);
23     else CheckBoxSelectedSet(&CheckBox4, 0);
24 }
25
26 void CheckBox6OnChange(tWidget *pWidget, unsigned long bSelected)
27 {
28
29 }
30

```

- b. Edit the code in the programming window using the C language. (Users can copy the code from the picture below.)

- c. When finished, click “OK” to store.



b.

```

HANDLE h;
char ReadDI_value[1];
int NetID = 1;
int addr = 0;
int ch_count = 8;
char DO_value[1];

void Frame12OnCreate()
{
    h = uart_Open("COM1,9600,N,8,1");
}

void Frame12OnDestroy()
{
    uart_Close(h);
}

void Timer8OnExecute(tWidget *pWidget)
{
    mrm_ReadDI (h, NetID, addr, ch_count, ReadDI_value);

    if (ReadDI_value[0] == 1) CheckBoxSelectedSet(&CheckBox4, 1);
    else CheckBoxSelectedSet(&CheckBox4, 0);
}

void CheckBox6OnChange(tWidget *pWidget, unsigned long bSelected)
{
    if(bSelected) DO_value[0] = 1;
    else DO_value[0] = 0;
    mrm_WriteDO (h, NetID, addr, ch_count, DO_value);
}

```

C.

Declare the related variables of DO.

Send the control message when the User clicks the button.

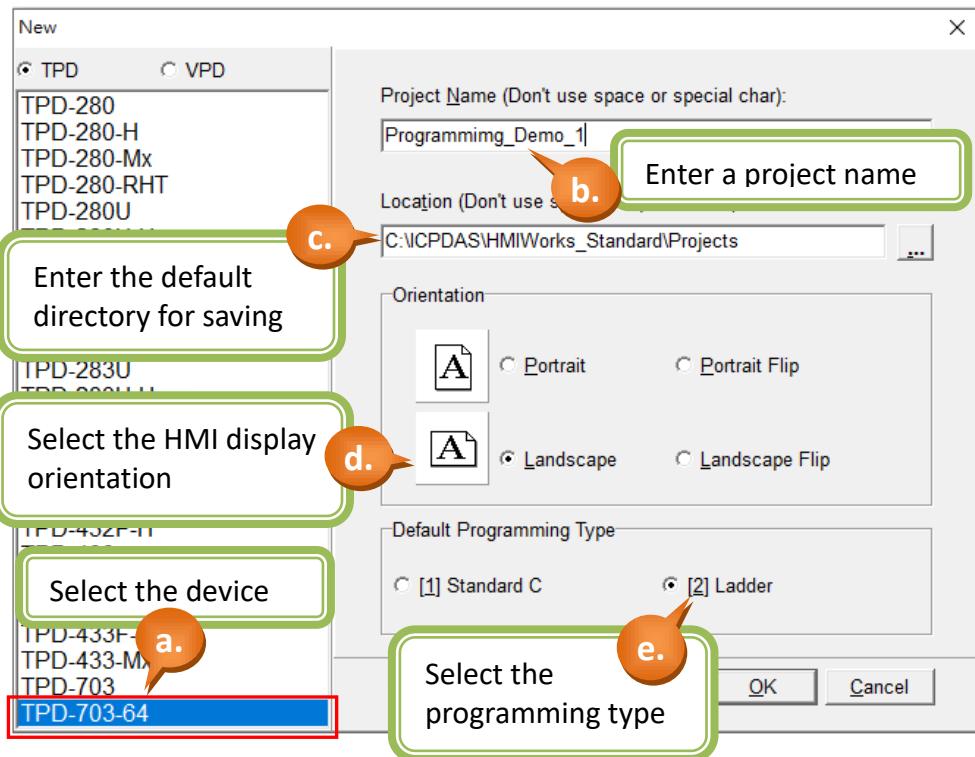
**Step 11: Build and Download the demo to the TouchPAD model.
Please refer [2.4 Downloading Methods for TouchPAD](#)**

3.2 Using the Ladder Designer

3.2.1 Demo 1 (Beep & Count)

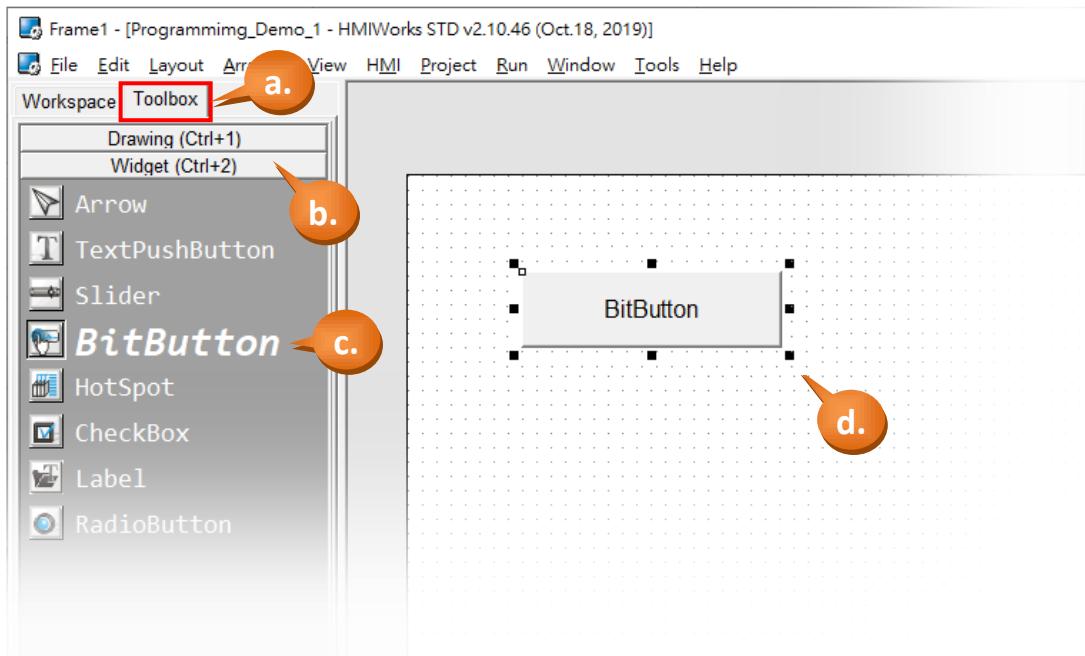
Demo 1 is programmed using the Ladder Designer and implements a single button that will beep when clicked on and will increase the counter by one. The counter will reset to zero when it reaches to ten.

Step 1: Create a new TPD-703-64 project based on the Ladder Designer (refer to [Section 2.2](#) for more details).

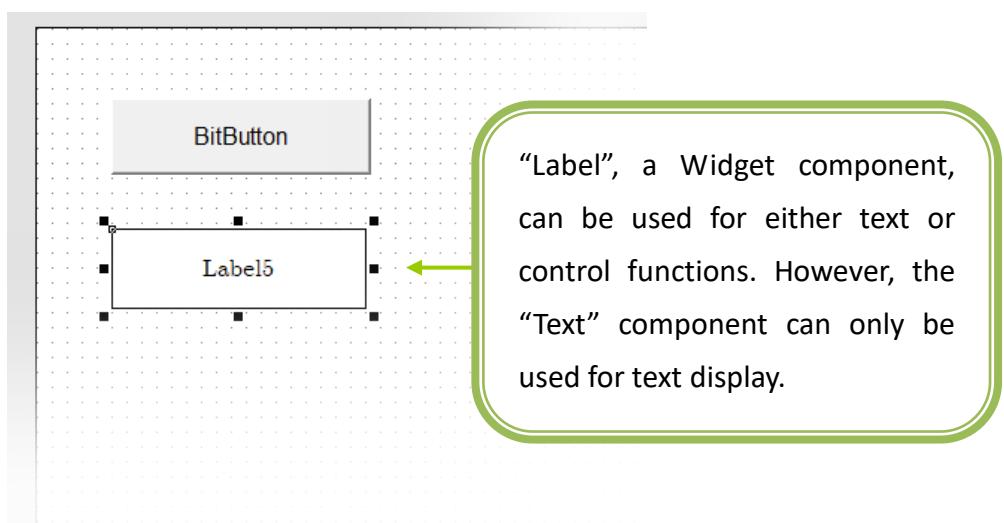


Step 2: Create a “BitButton” (picture button).

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “Widget (Ctrl+2)” heading to unfold the “Toolbox” panel.
- c. Select the “BitButton” icon, then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click and drag the mouse to draw a rectangle.

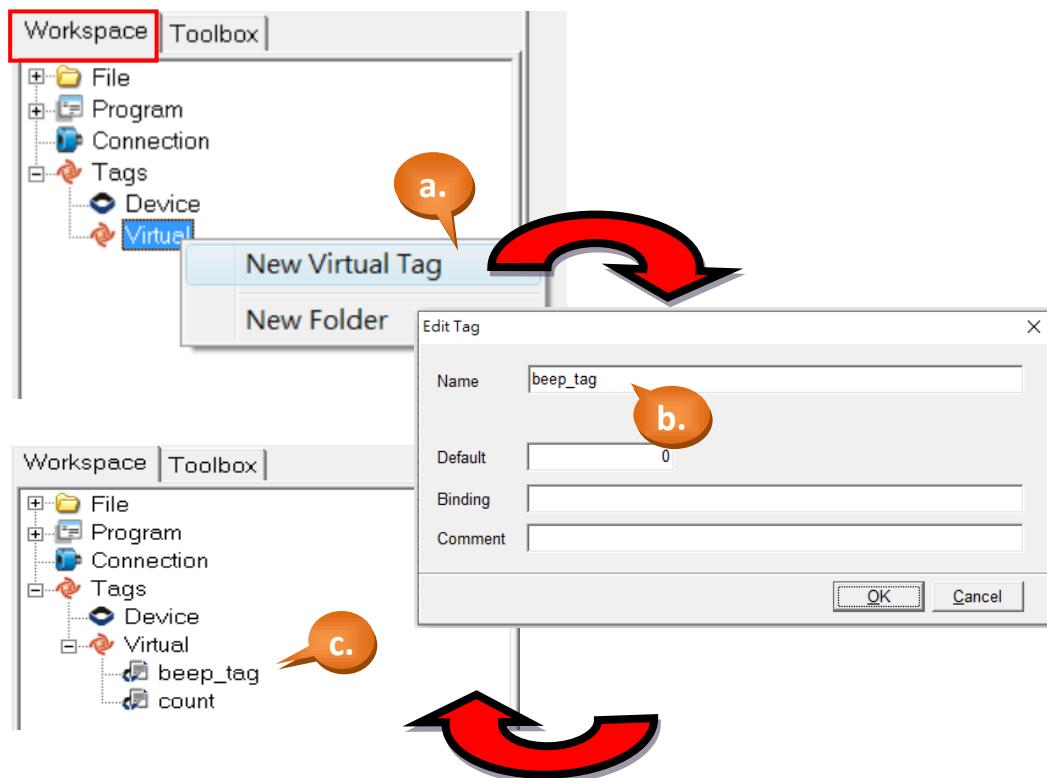


Step 3: In the same manner as the previous step, click on the “Label” icon and create a “Label” (text display) in the design area.



Step 4: Create two tags, “beep_tag” and “count”.

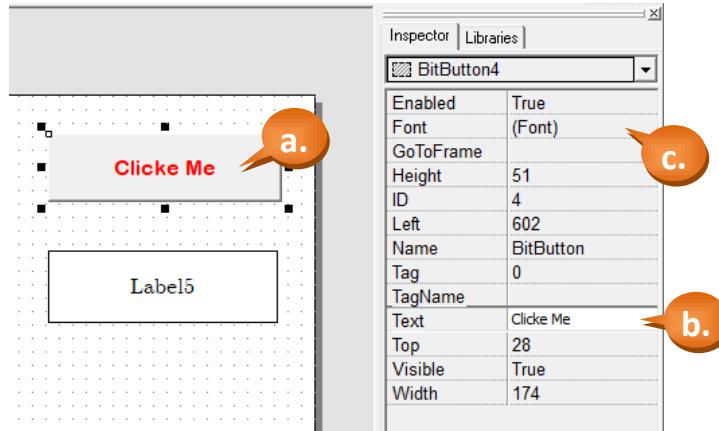
- a. As shown in the picture below, right click the “Virtual” in the Workspace to create the tags.
- b. Enter the name of the tag in the “Edit variable” window. Here we use “beep_tag” for the first tag and “count” for the second.
- c. Finally, we can see that the 2 tags created in the Workspace.



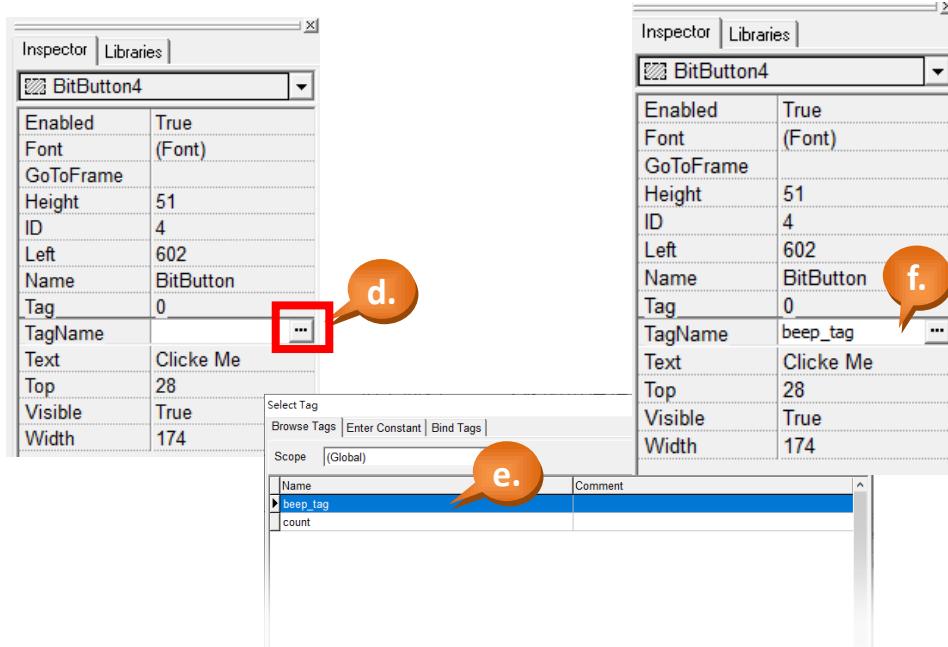
NOTE : The Virtual Tag like as a integer of C language, it can record the range -2147483648 to 2147483647.

Step 5: Configure the properties for the “BitButton” component.

- Select the “BitButton” component in the design area.
- Click the “Text” field in the Properties Inspector panel and enter the string “Click Me”.
- Click the “Font” field to change the font to a suitable style, size and color.

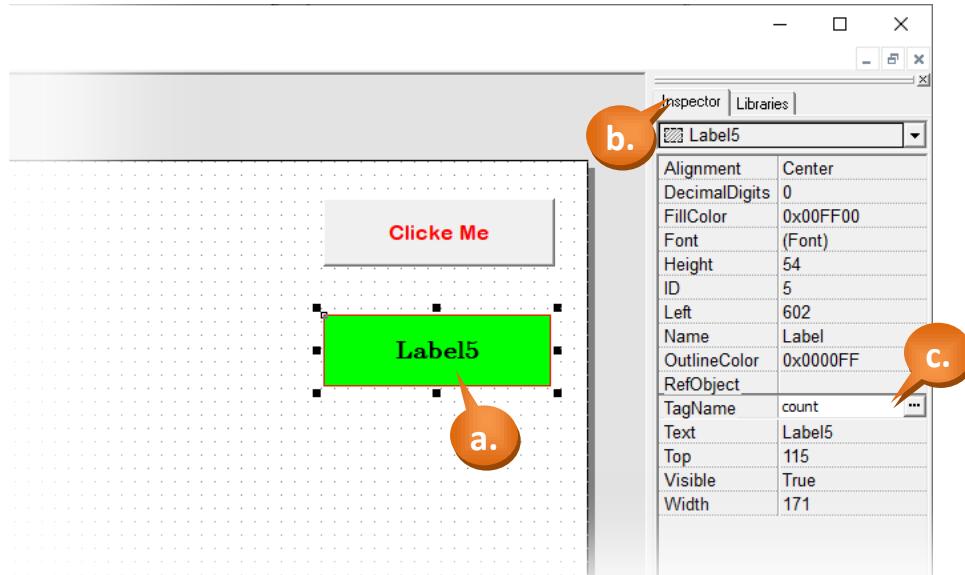


- Click on the “TagName” property in the Inspector and there shows a little button “...”.
- Click on the “...” button.
- Select the tag you want. Here we select the tag, “beep_tag”, for the BitButton.
- Finally, we can see the tag we just choose.



Step 6: Configure the properties for the “Label” component.

- a. Select the Label component in the design area.



- b. The following properties can be modified from the Properties Inspector area on the right hand side:

Alignment: Sets the text alignment position in the display box of the “Label” component.

DecimalDigits: The position of decimal point.

FillColor: Sets the fill color for the display box of the “Label” component.

Font: Sets the font for the text in the display box.

Height: Sets the height of the display box.

ID: A unique ID number to identify different entities of the same type in the design.

Left: Sets the X-coordinate of the top left corner of the display box.

Name: The type name of the component.

OutlineColor: Sets the outline color of the display box.

RefObject: Let the outward refer the picture in the ObjectList Widget.

TagName: Let the value of widget refer to the specified tag.

Text: Sets the text to be displayed in the display box.

Top: Sets the Y-coordinate of the top left corner of the display box.

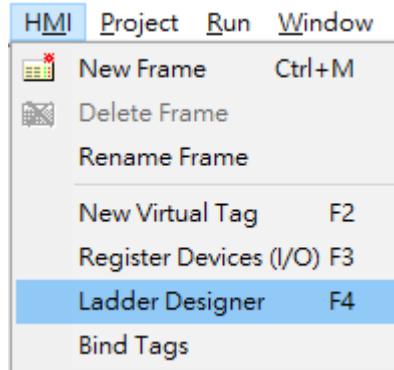
Visible: Sets the visibility of this Widget on the screen of model.

Width: Sets the width of the display box.

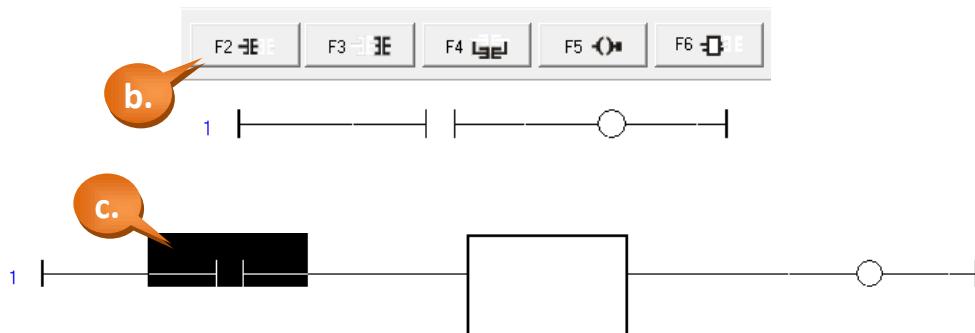
- c. Select “TagName” as “count”. (It is similar to set “TagName” as the above step.)

Step 7: Use the Ladder Designer to program the logic – the first rung.

- Click the menu [HMI] > [Ladder Designer].



- Press the F2 button (or F2 key on the keyboard) to create a new rung.
- Move the cursor (the highlighted rectangle) to the first contact symbol and then press F7 to add a new function block to the right of it.



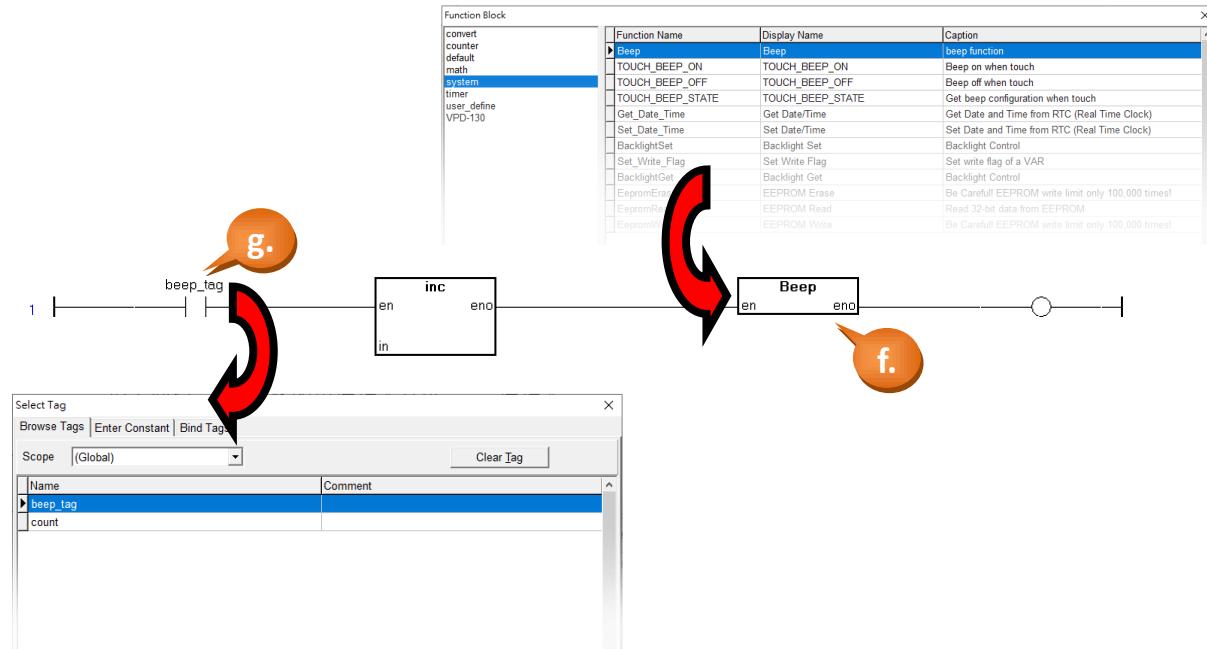
- Double-click on the function block to set the function to it. Here we set it the function, "increment", of the category, "math".

Function Block			
	Function Name	Display Name	Caption
convert	Add	+	add function
counter	Sub	-	sub function
default	Mul	*	mul function
math	Div	/	div function
system			
timer			
user_define			
VPD-130			
	Inc	inc	increment function
	dec	dec	decrement function
	Mod	%	mod number
	Scaling	Scaling	Scaling
	ScalingS16	ScalingS16	Scaling (Short16)
	Invert01	Invert01	Invert 0 and 1
	Exp	Exp	Natural exponential function
	Ln	Ln	Natural logarithmic function (base e)
	Log	Log10	Logarithmic function (base 10)

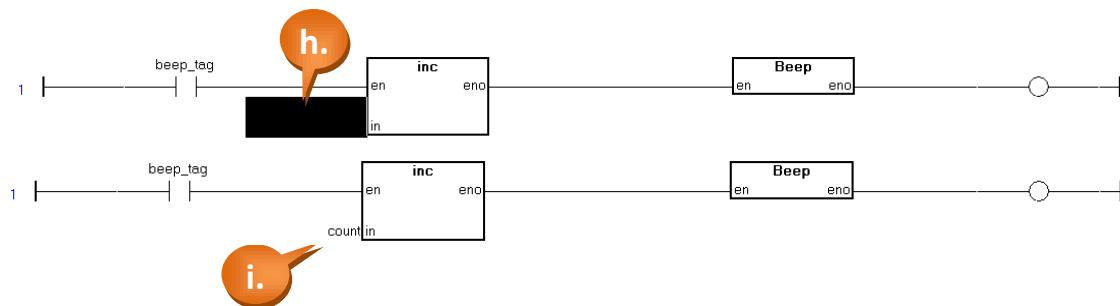
- Move the cursor to the function block of "inc" and then press the F7 button again to add a new function block to the right of it.



- f. Double-click on the second function block to set the function to it. Here we set it the function, “Beep”, of the category, “system”.
- g. Double-click on the contact symbol to select variable “beep_tag”.



- h. Similarly, double-click on the neighborhood of the input parameter “in” as below figure to select variable “count”.
- i. Finally, the first rung is done.



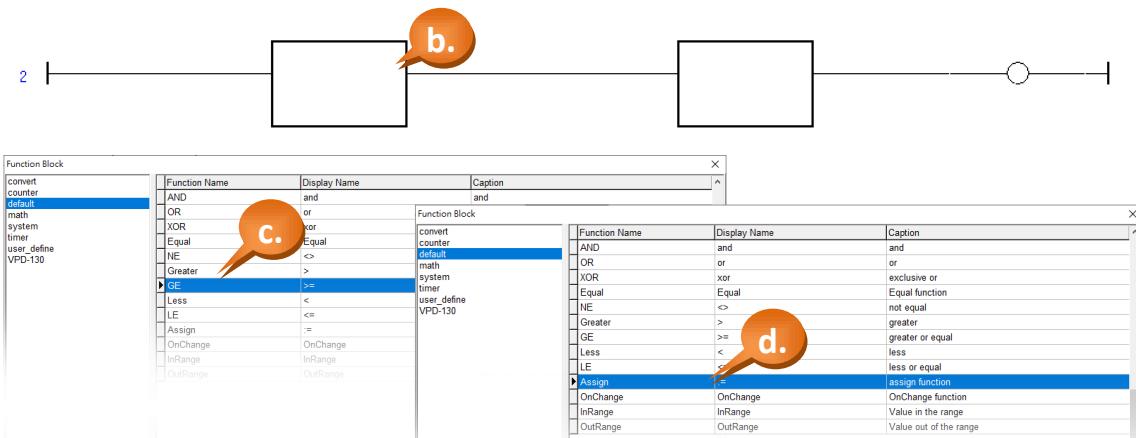
- j. The first rung does the followings:
- ```

if(beep_tag == 1) //that is, when the BitButton is pressed.
{
 count = count + 1; //increment the count
 Beep(); //make a beep sound
}

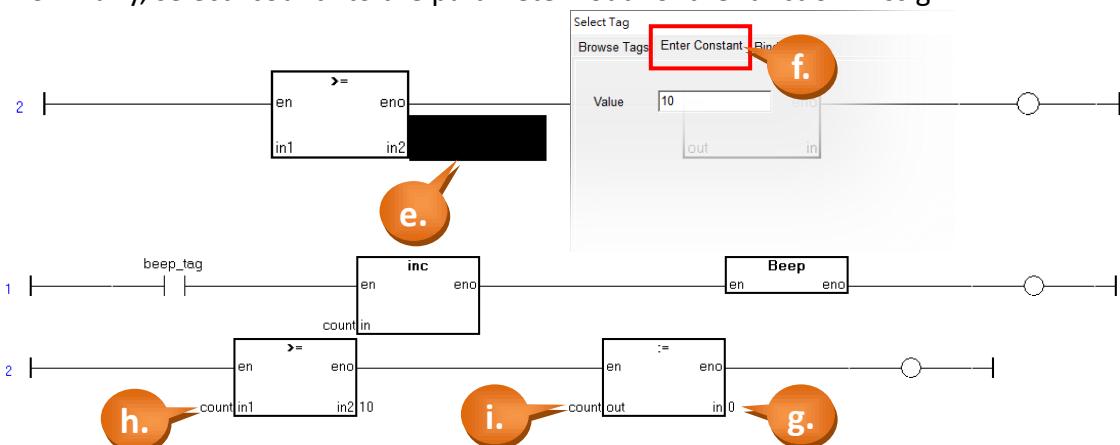
```

**Step 8: Use the Ladder Designer to program the logic – the second rung.**

- Similar to Step 7. Move the cursor to an empty position and then press the F6 button to add a new rung with a function block.
- Move the cursor to the previously-created function block and then press the F7 button to create a new function block to the right of the first one.
- Double-click on the first function block to set its function to “GE” (greater than or equal to) of the category, “default”.
- Double-click on the second function block to set its function to “Assign” of the category, “default”.



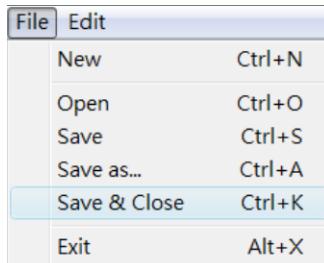
- Set the value of the parameter “in2” of the function “GE” ( $\geq$ ) by double-clicking on the neighborhood of the parameter “in2”.
- To enter a constant to a parameter, click on the tab “Enter Constant” in the “Select variable” window. Here we enter 10.
- Similarly, set the parameter “in” of the function “Assign” to zero.
- Similarly, select “count” to parameter “in1” of the function “GE” ( $\geq$ ).
- Similarly, select “count” to the parameter “out” of the function “Assign”.



- j. The second rung does the followings:

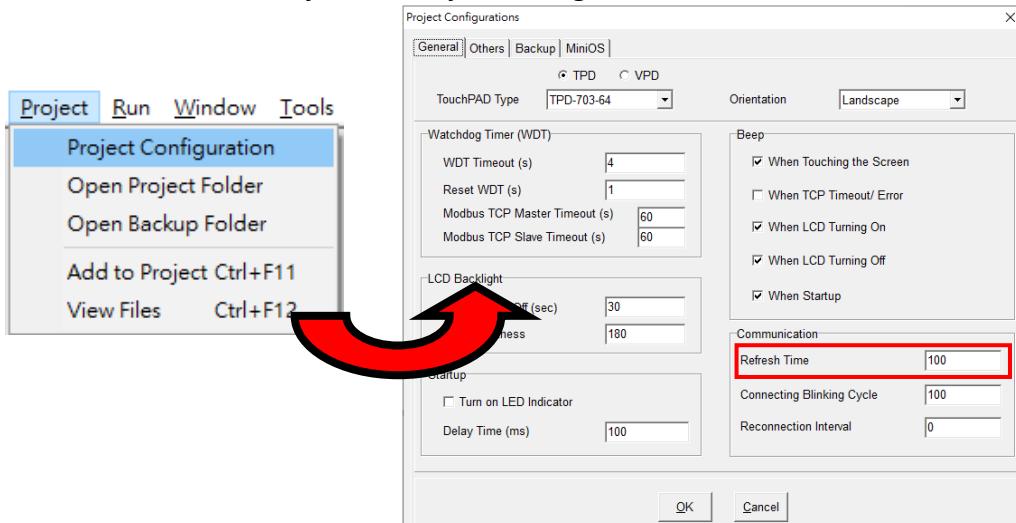
```
if(count >= 10)
{
 count = 0; //set the count to zero
}
```

**Step 9: Save and Close (the Ladder).**



**Step 10: Tune the scan time if necessary.**

- a. Click the menu [Project] > [Project Configuration] to set the scan time.



- b. The default scan time is 100 ms. The scan time is the time interval between two consecutive Ladder scan. That is, TouchPAD executes each rung of the ladder serially from the first rung to the last and after finishing the last rung, it waits until the scan time interval is up (the scan time interval is calculated from the time of the first rung's execution) to execute from the beginning (the first rung) again.
- c. The scan time might influence the feeling of sensitivity. If the scan time is too small, touching the BitButton on the screen one time may causes it triggering many times and thus makes the counter increment more than one.

**Step 11: Build and Download the demo to the TouchPAD model.**

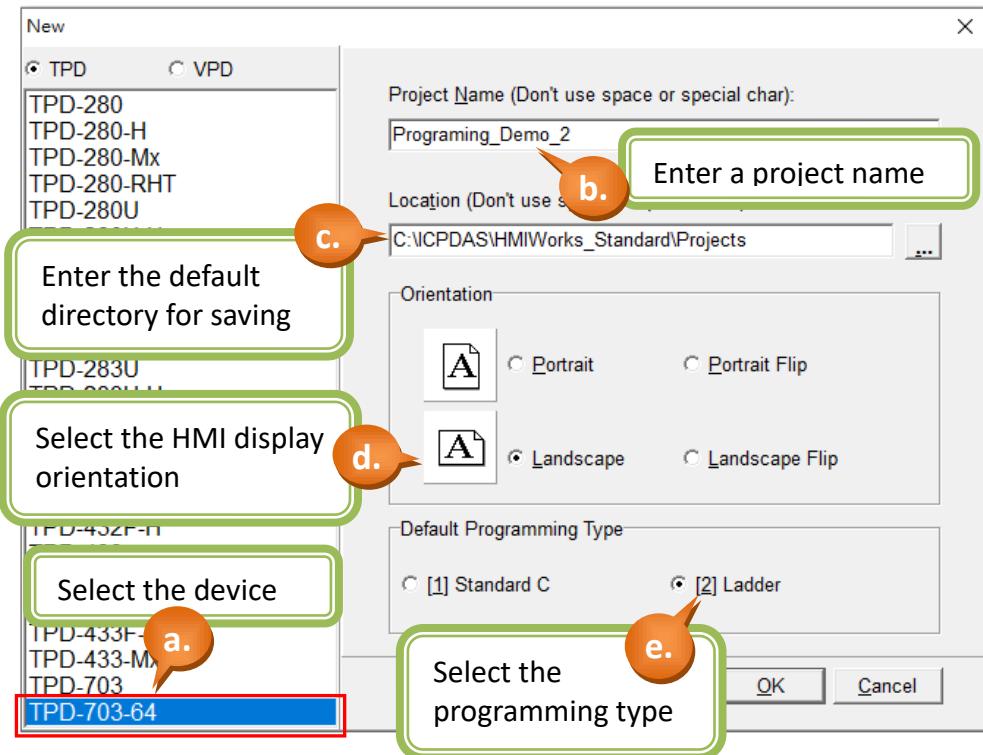
Please refer [2.4 Downloading Methods for TouchPAD](#)

### 3.2.2 Demo 2 (Displaying the Decimal Point)

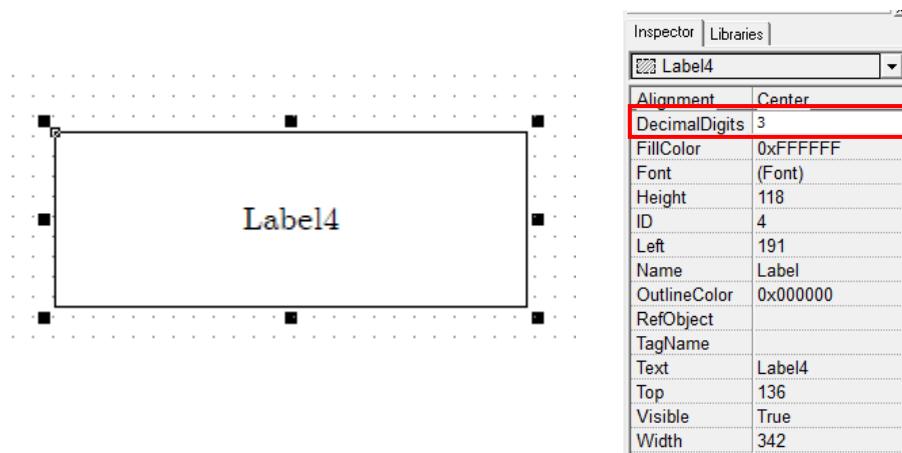
The numbers used in the “Ladder Designer” are displayed in integer format without decimal digit. In some case, you may need to calculate or display the decimal. Please magnify the value enough to calculate it with integer type, and then display the result using the “DecimalDigits” in the Property Inspector. The detail is below:

Suppose we want to calculate the result of “50 divided by 7”, and the precision requires the third decimal place (the fourth digit is chopped off). We need magnify the “50” to “50000” and dividing it by 7, and then adjust the “DecimalDigits” of the Label to “3” to display it.

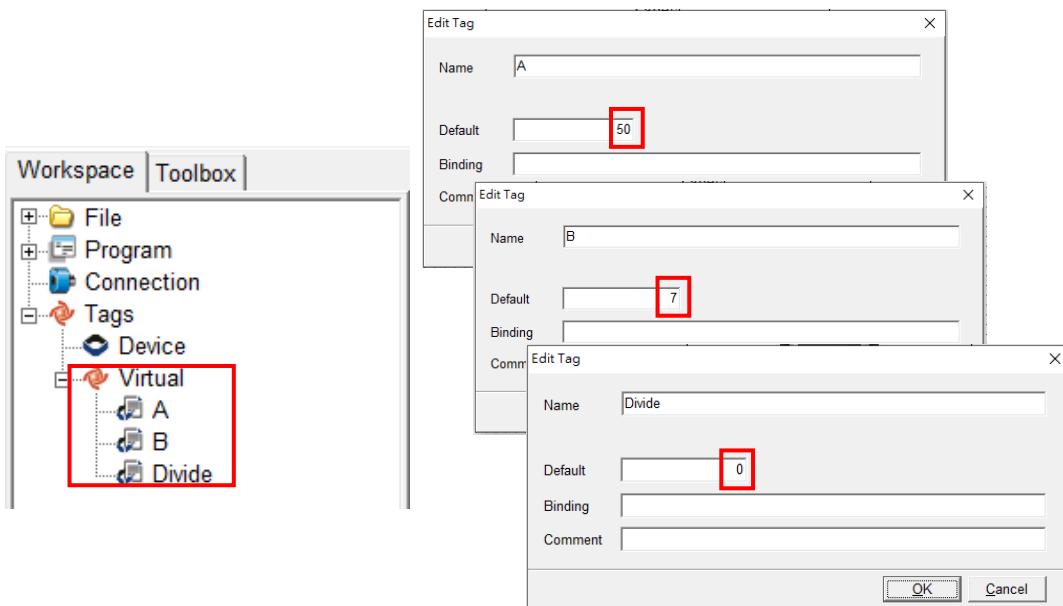
Step 1: Create a new TPD-703-64 project based on the Ladder Designer (refer to [Section 2.2](#) for more details).



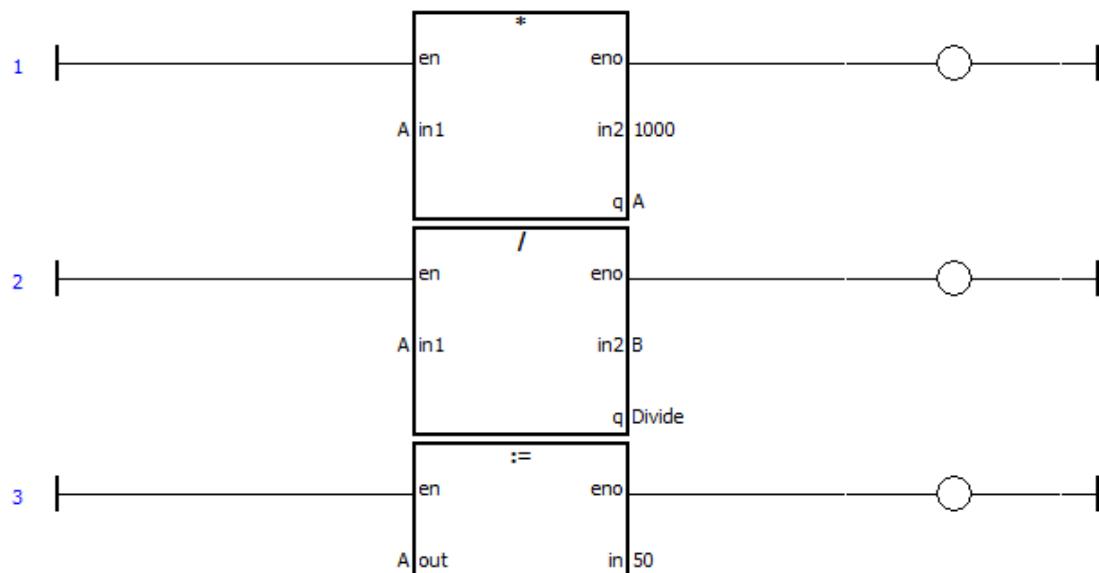
Step 2: Create a Label and adjust its “DecimalDigits” property to “3”.



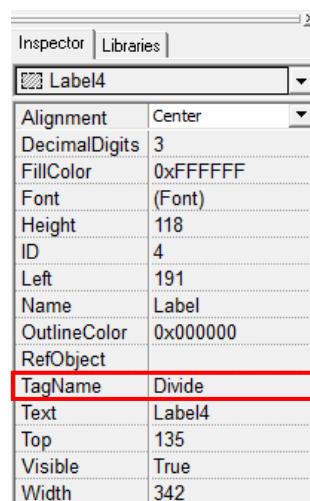
Step 3: Create two Virtual Tags, i.e., A and B, which have the default value “50” and “7”, respectively. And then create a Virtual Tag, i.e., Divide, which have the default value “0”.



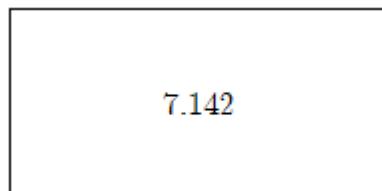
Step 4: Use the Ladder Designer to program the logic.



Step 5: Let the “TagName” of Label to select the “Divide” tag.



Step 6: Build and Download the demo to the TouchPAD model.  
Please refer [2.4 Downloading Methods for TouchPAD](#)



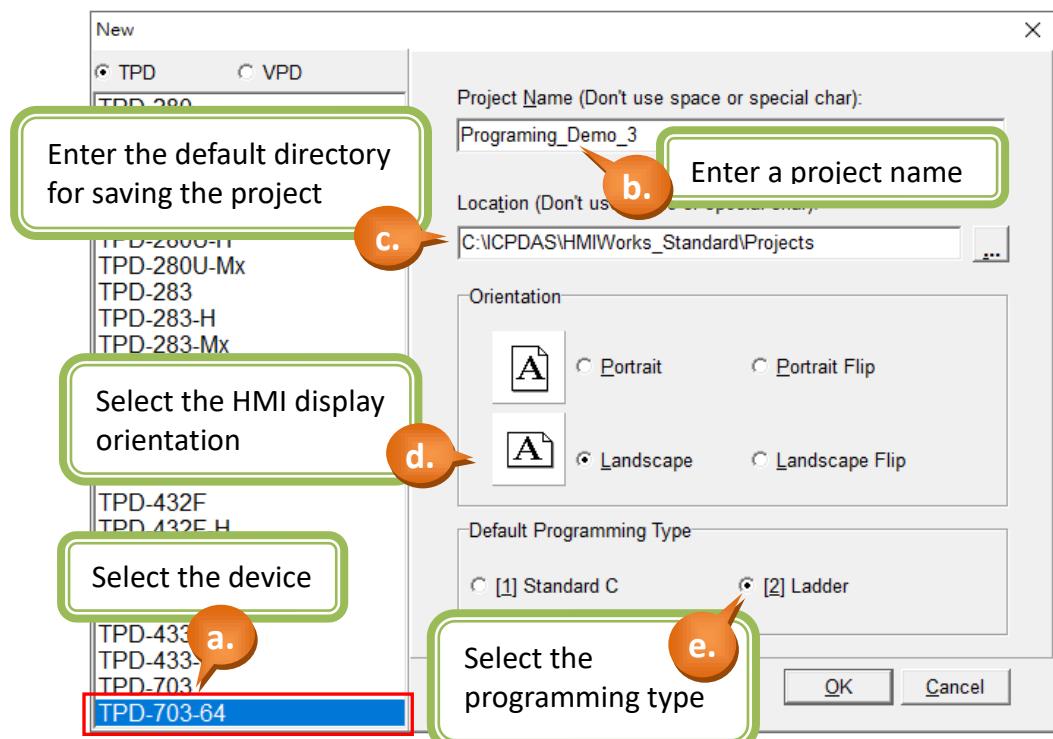
### 3.2.3 Demo 3 (Digital Clock)

**Demo 3** is programmed using the Ladder Designer and implements a digital clock that will update the real time using built-in RTC. And this demo shows how to add the background which is designed by user.



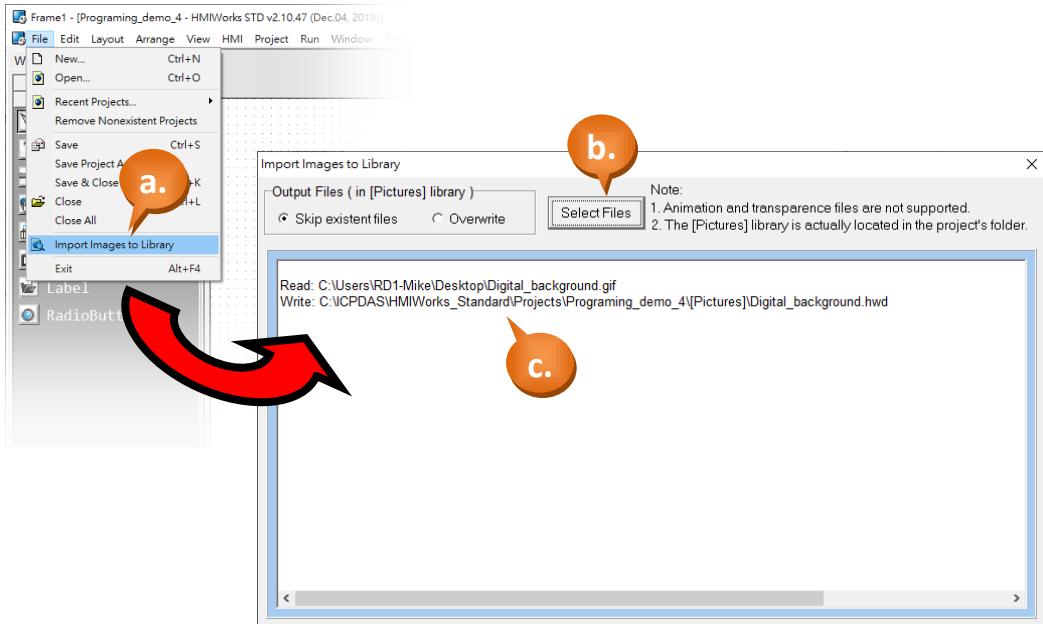
NOTE : Demo 3 use the built-in RTC in the TPD/VPD model, the TPD-280-H/Mx and TPD-283-H/Mx model are not support the RTC function.

Step 1: Create a new TPD-703-64 project based on the Ladder Designer (refer to [Section 2.2](#) for more details).



## Step 2: Import a picture as the background.

- a. Select the “Import Images to Library” in the “File” menu to open the dialog box.
- b. Click the “Select Files” button and choice the picture, i.e., the background.
- c. After choosing the picture, the dialog box will show the path of related file, and that means the importation is completed.

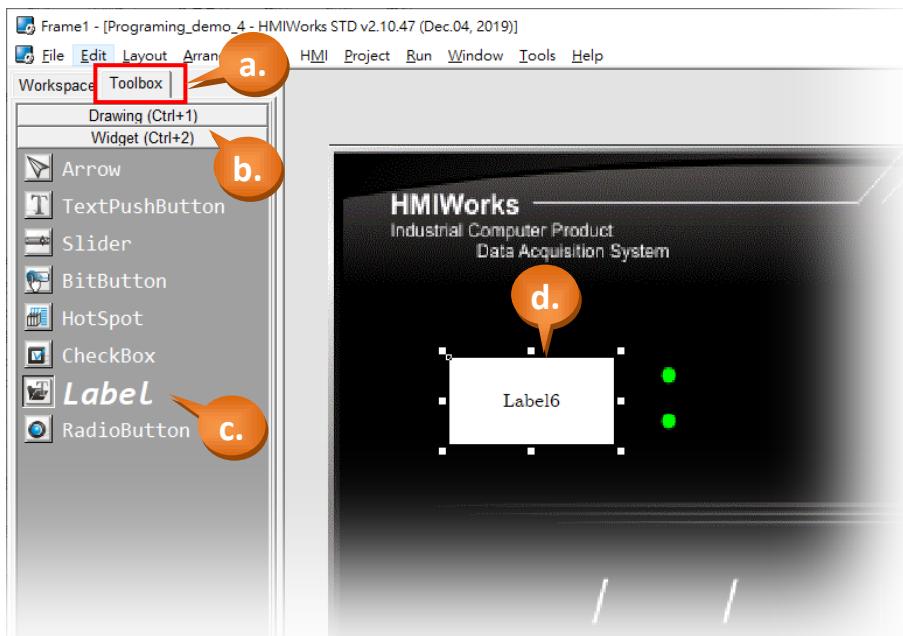


- d. Close the “Import Images to Library” dialog box and click the “Libraries”
- e. Select the “Pictures” category and you will find the imported picture.
- f. Select and drag the imported picture to the design area.

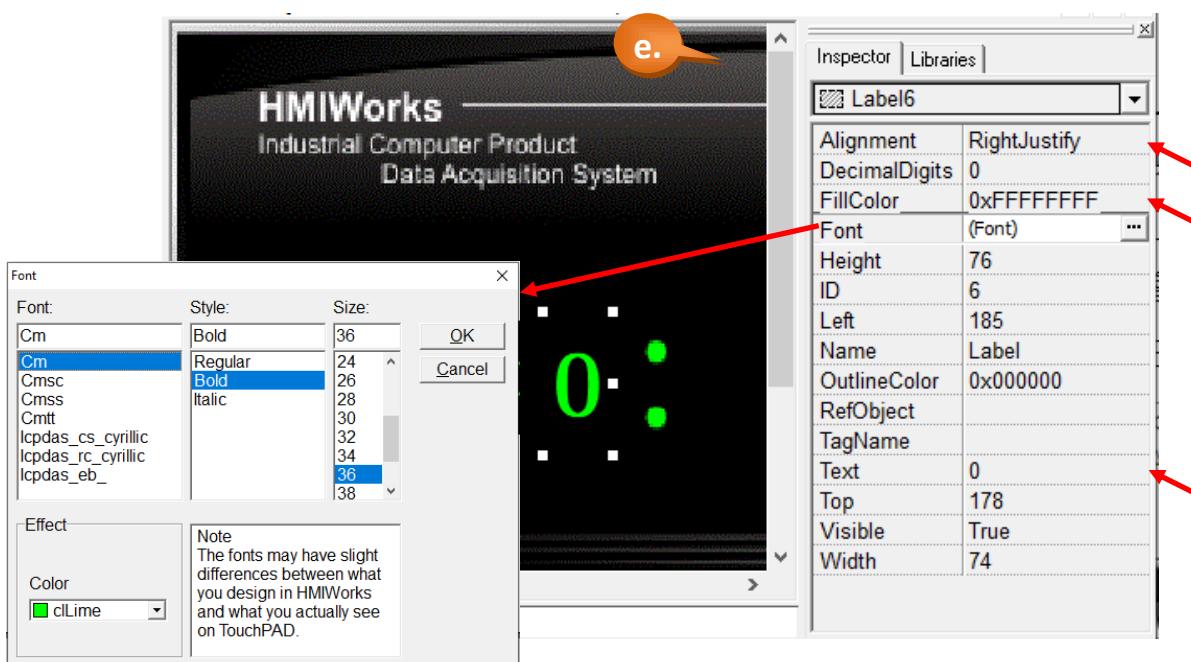


Step 3: Create a “Label” (text display).

- a. Switch to the “Toolbox” tab, as shown in the picture below.
- b. Click the “Widget (Ctrl+2)” heading to unfold the “Toolbox” panel.
- c. Select the “Label” icon, then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- d. In the design area, click and drag the mouse to draw a rectangle.

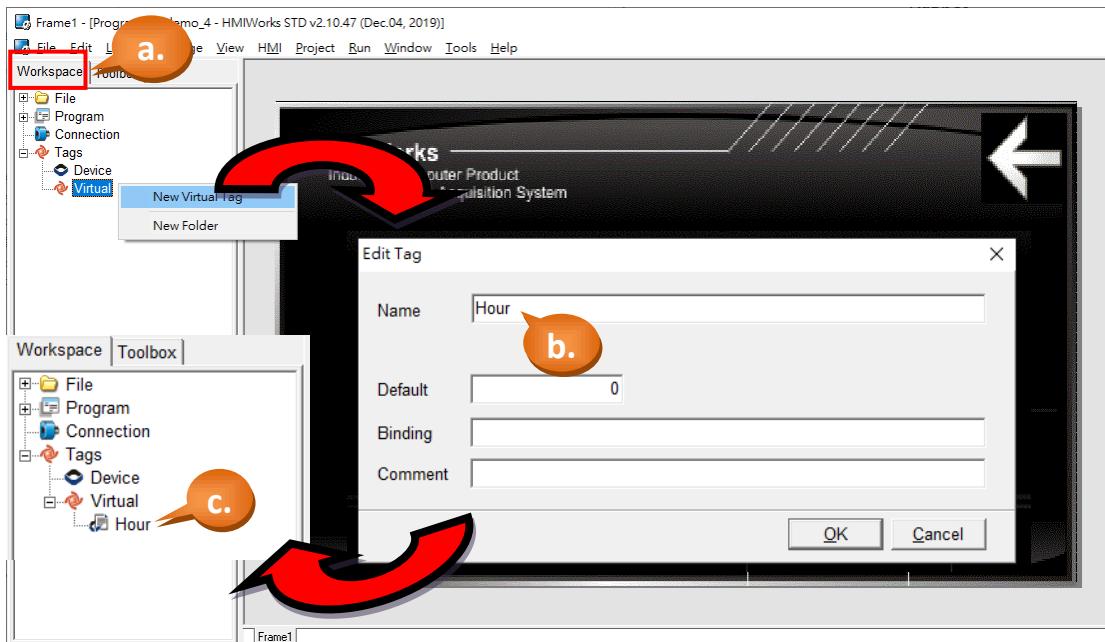


- e. Click the “Alignment”, the “FillColor”, the “Font” and the “Text” field to change the style from the Property Inspector, as shown in the picture below.

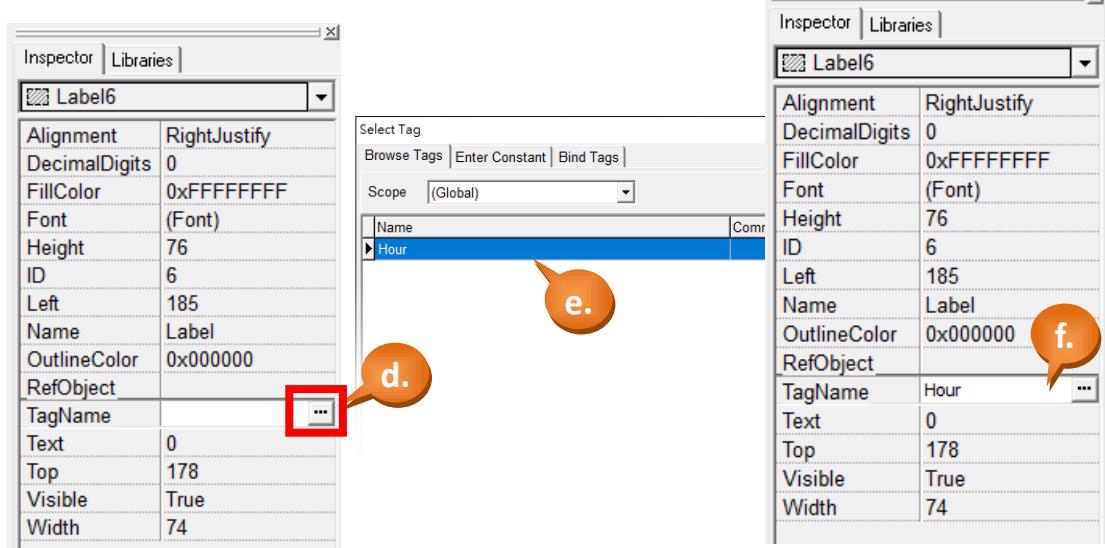


#### Step 4: Create the corresponding Virtual Tag.

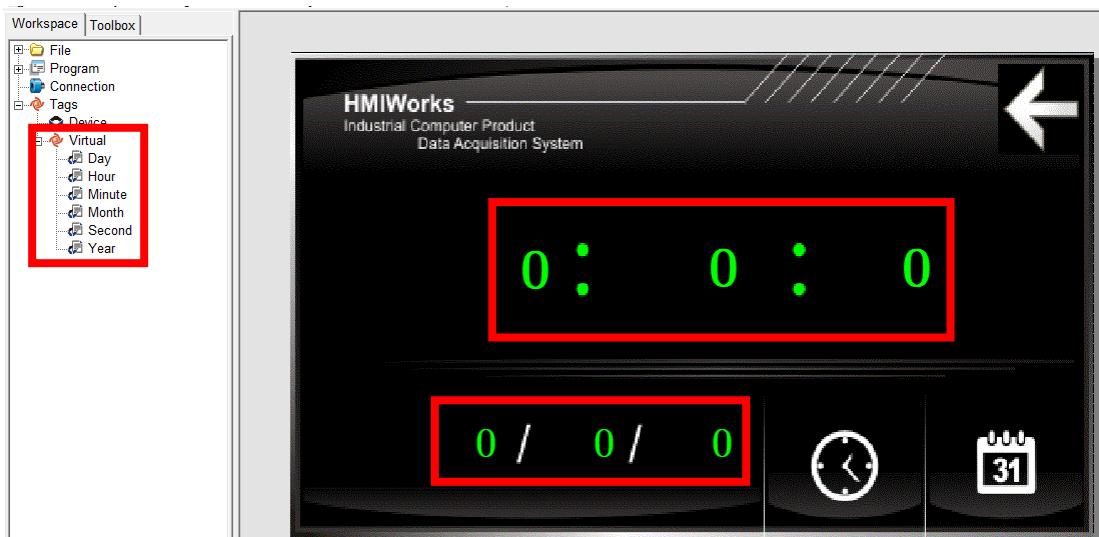
- As shown in the picture below, right click the “Virtual” in the Workspace to create the tags.
- Enter the name of the tag in the “Edit variable” window. Here we use “Hour”.
- Finally, we can see that the tag created in the Workspace.



- Click the “Label6” and then click the “TagName” property in the Inspector and there shows a little button “...”. Click on the “...” button.
- Select the tag you want. Here we select the tag, “Hour”.
- Finally, we can see the tag we just choose.

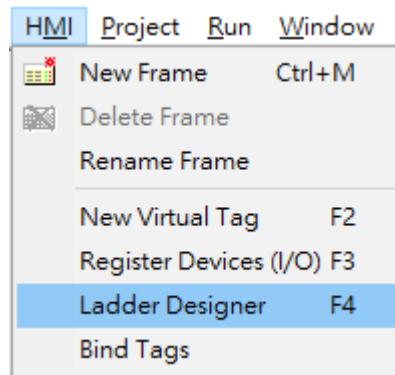


Step 5: Repeat the step 3 and step 4 to create others Label and the corresponding Virtual Tag.



Step 6: Use the Ladder Designer to program the logic.

- Click the menu [HMI] > [Ladder Designer].



- Press the F7 button (or F7 key on the keyboard) to create a new function block.

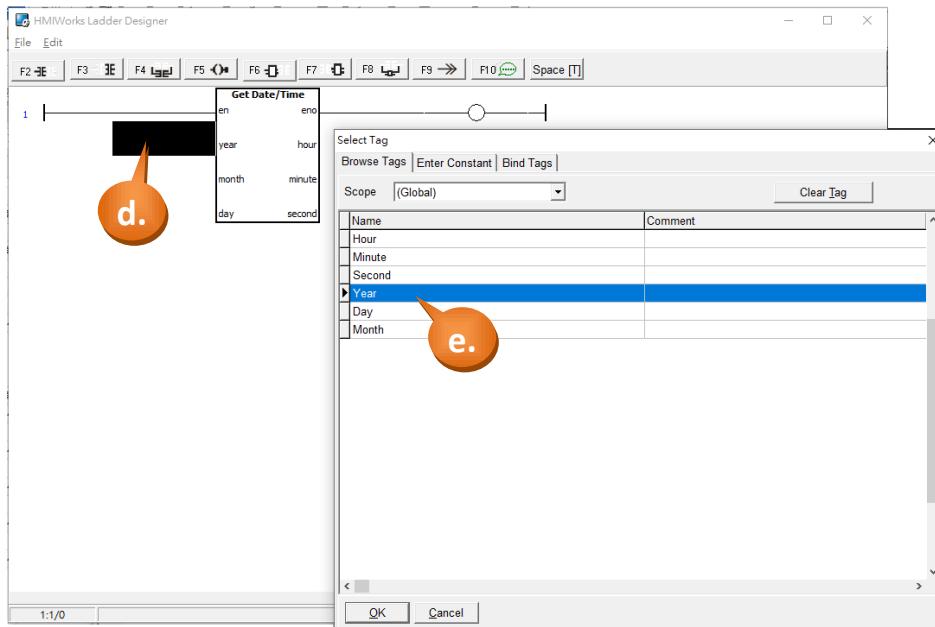


- c. Double-click on the function block to set the function to it. Here we set it the function, “Get\_Date\_Time”, of the category, “system”.

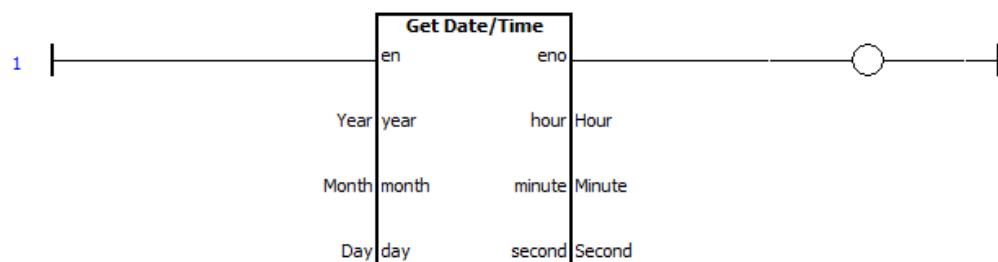
| Function Block       |                      |                                                     |
|----------------------|----------------------|-----------------------------------------------------|
| Function Name        | Display Name         | Caption                                             |
| convert              |                      |                                                     |
| counter              |                      |                                                     |
| default              |                      |                                                     |
| math                 |                      |                                                     |
| <b>system</b>        |                      |                                                     |
| timer                |                      |                                                     |
| user_define          |                      |                                                     |
| VPD-130              |                      |                                                     |
| <b>Get_Date_Time</b> | <b>Get Date/Time</b> | <b>Get Date and Time from RTC (Real Time Clock)</b> |
| Set_Date_Time        | Set Date/Time        | Set Date and Time from RTC (Real Time Clock)        |
| BacklightSet         | Backlight Set        | Backlight Control                                   |
| Set_Write_Flag       | Set Write Flag       | Set write flag of a VAR                             |
| BacklightGet         | Backlight Get        | Backlight Control                                   |
| EepromErase          | EEPROM Erase         | Be Careful! EEPROM write limit only 100,000 times!  |
| EepromRead           | EEPROM Read          | Read 32-bit data from EEPROM                        |
| EepromWrite          | EEPROM Write         | Be Careful! EEPROM write limit only 100,000 times!  |

- d. Double-click on the neighborhood of the input parameter “year” as below figure.

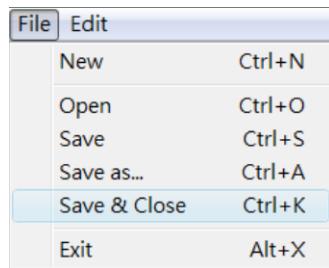
- e. Select variable “Year” to store the output of “Get\_Date\_Time” block.



- f. Repeat these steps to let all Virtual Tag correspond to the outputs of “Get\_Date\_Time” block, as shown in the picture below.



Step 7: Save and Close (the Ladder).



Step 8: Build and Download the demo to the TouchPAD model.  
Please refer [2.4 Downloading Methods for TouchPAD](#)



### 3.2.4 Demo 4 (Remotely control the ICPDAS I/O model)

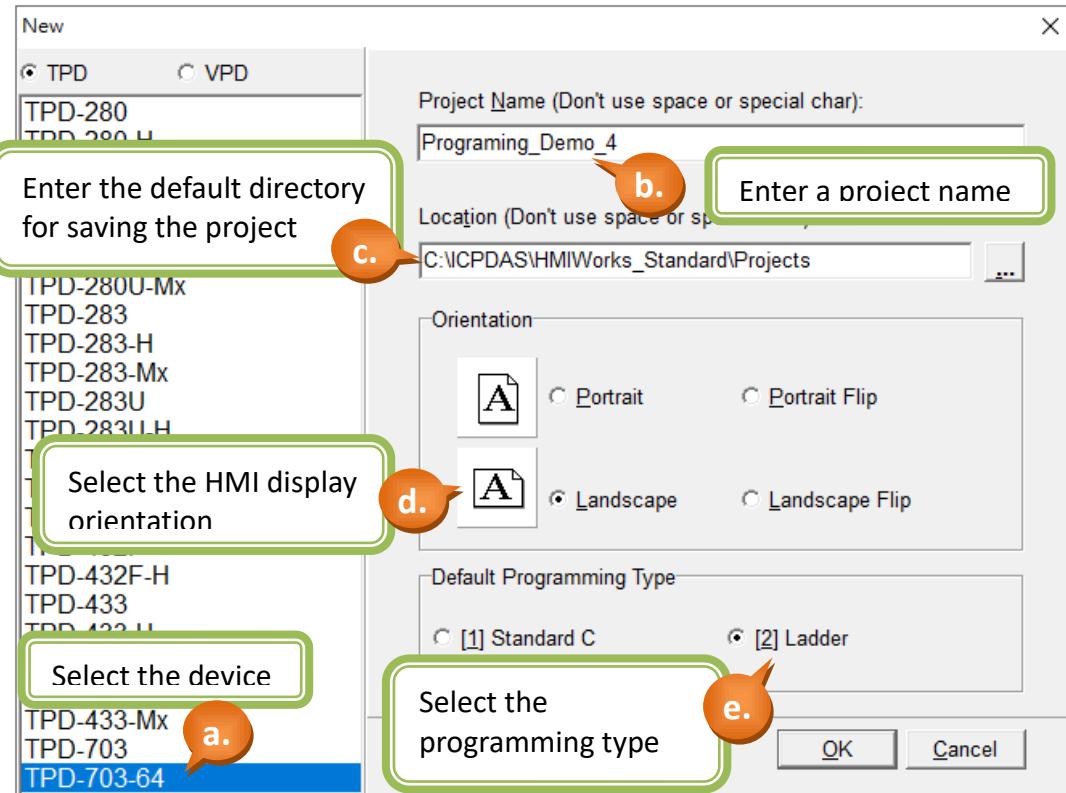
**Demo 4** is programmed using the Ladder Designer and implements that remotely control the ICPDAS I/O model.

In this demo, we use the ICPDAS M-7055 model to demonstrate. This demo will use the switch on the screen of the TPD model to control the DO of the M-7055 model, and the TPD also update the DI status of the M-7055 model at any time on its screen.

Please connect the DI pin and the DO pin of the M-7055 model, i.e., DI0 connects to DO0, DI1 connects to DO1, DI2 connects to DO2... etc. Please refer the manual of M-7055 to view the detailed wiring.

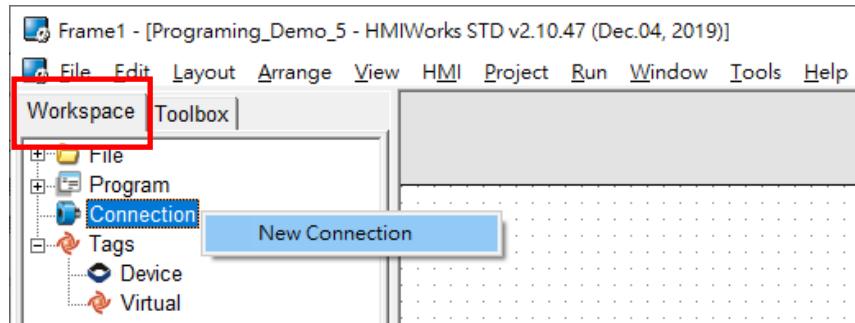
[http://www.icpdas.com/web/product/download/io\\_and\\_unit/rs-485/document/manual/7000/I-7000\\_M-7000\\_DIO\\_en.pdf](http://www.icpdas.com/web/product/download/io_and_unit/rs-485/document/manual/7000/I-7000_M-7000_DIO_en.pdf)

Step 1: Create a new TPD-703-64 project based on the Ladder Designer (refer to [Section 2.2](#) for more details).

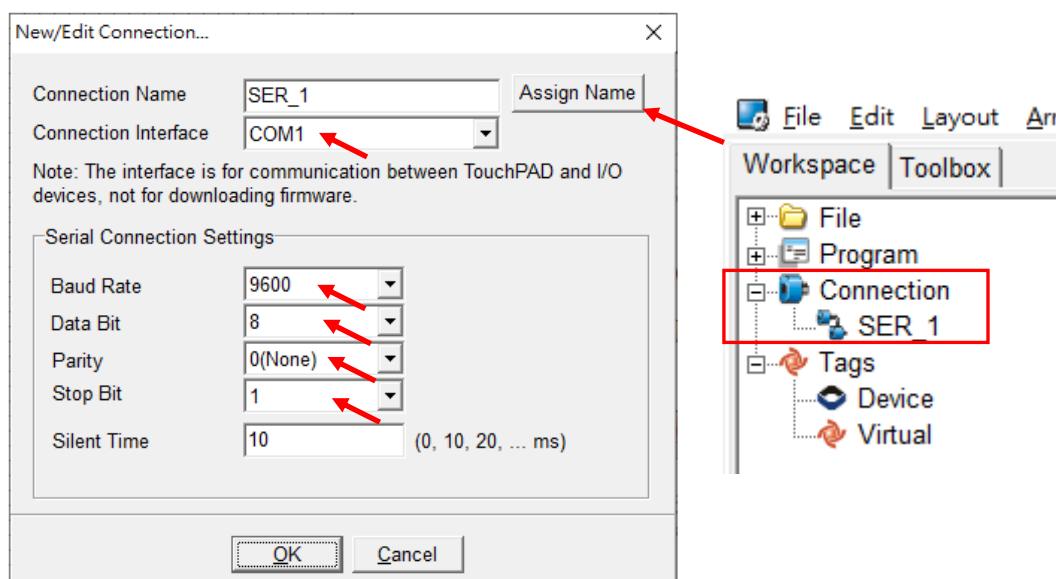


Step 2: Set up the connection.

- a. Right click the “Connection” item in the “Workspace” tab and select the “New Connection” option to set up the connection.

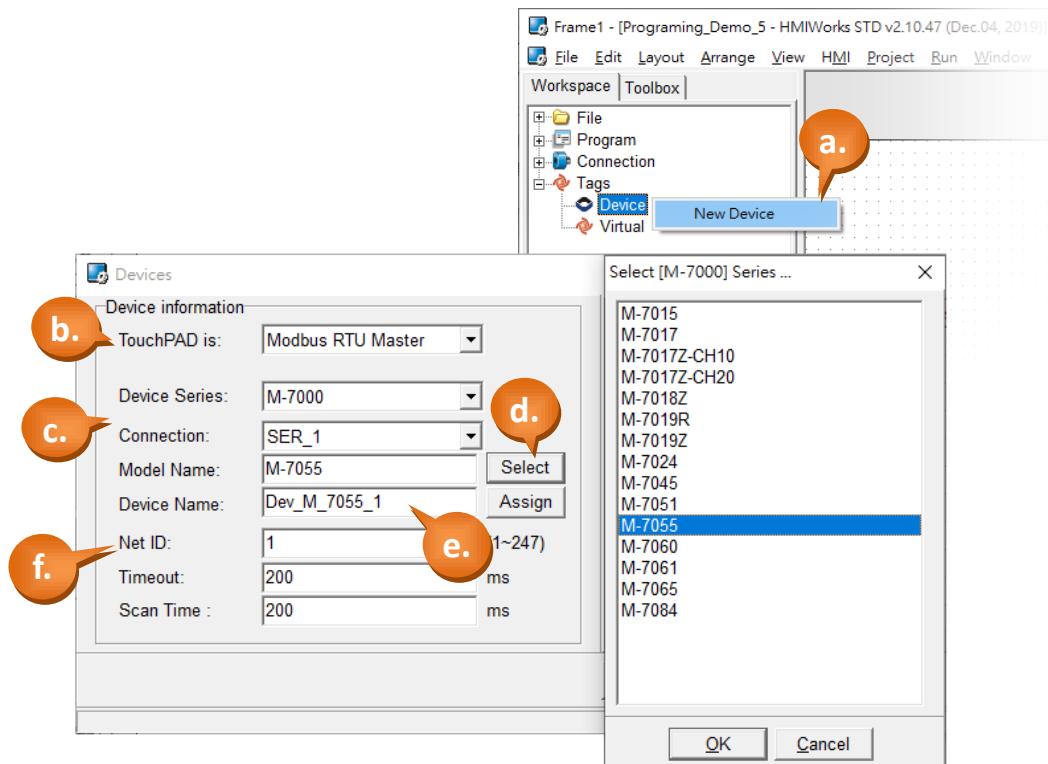


- b. For this demo, Select the “COM1” interface and set the “Baud Rate”, the “Data Bit”, the “Parity” and the “Stop Bit” to be the same as the M-7055 model (The settings of below picture is the default settings of the M-7055 model). And providing a connection name or click the “Assign name” button to have a name assigned by the system.



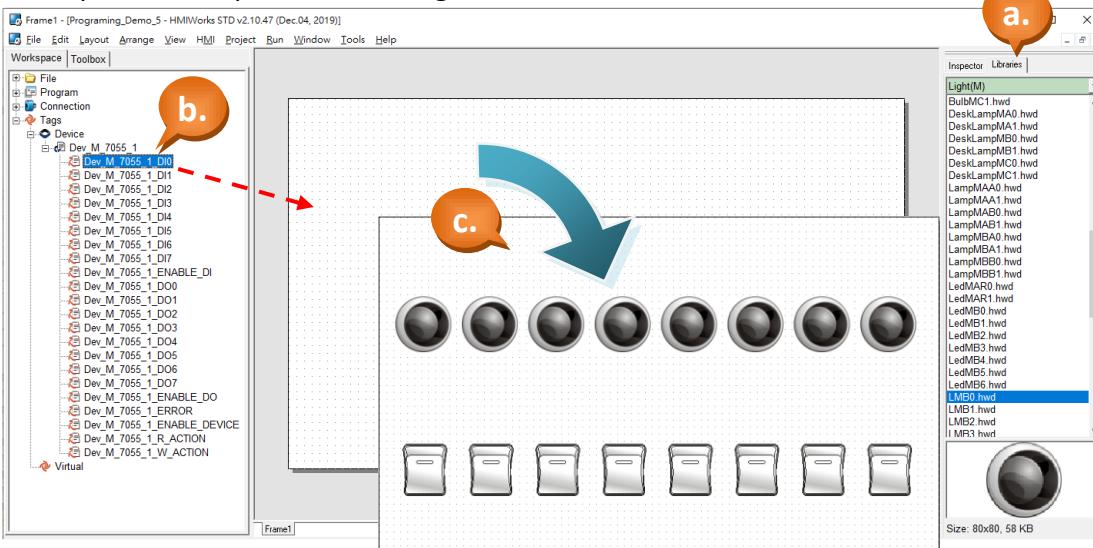
**Step 3: Import the Device tag of the I/O model.**

- a. Right click the “Device” item in the “Workspace” tab, then select the “New Device” option.
- b. Select “Modbus RTU Master” from the “TouchPAD is” dropdown menu.
- c. Select “M-7000” from the “Device Series” dropdown menu and then select the created “SER\_1” from the “Connection” dropdown menu.
- d. Click the “Select” button to select the I/O module. (This demo uses M-7055)
- e. Enter a connection name to the “Device Name” field or click the “Assign name” button to have a name assigned by the system.
- f. Input the address number of the M-7055 module in the “Net ID” field. (This demo uses **Net ID = 1**.



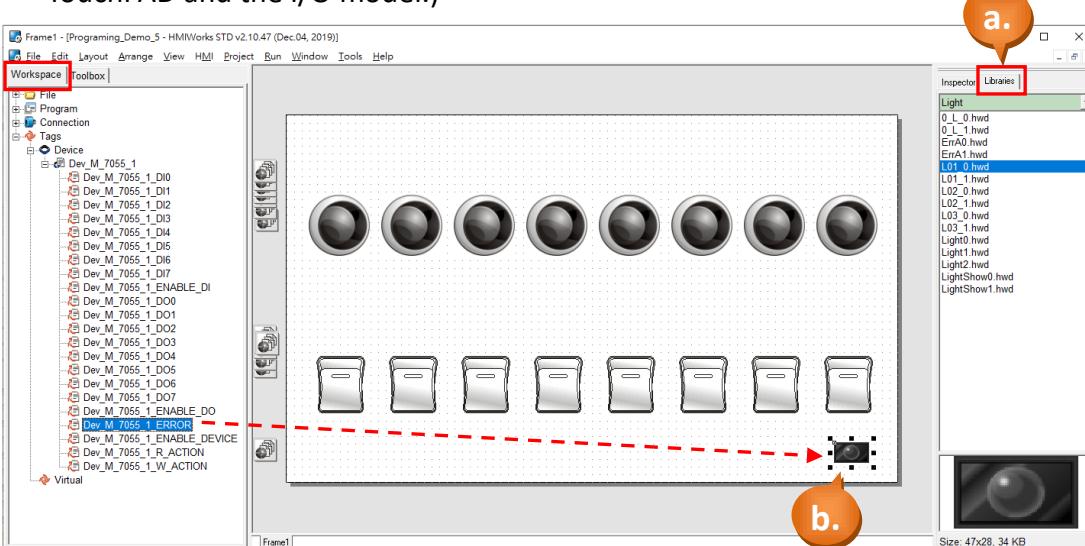
**Step 4:** Create the display LED and the control button. (This demo uses eight DI and DO respectively)

- Click the “Libraries” tab on the right hand side, and select the type of [graphic \(GUI\)](#) you want to use.
- Select and drag the corresponding I/O tag from the left hand side into the frame.
- Repeat the steps above to drag DI0 ~7 and DO0 ~7 into the HMI frame.



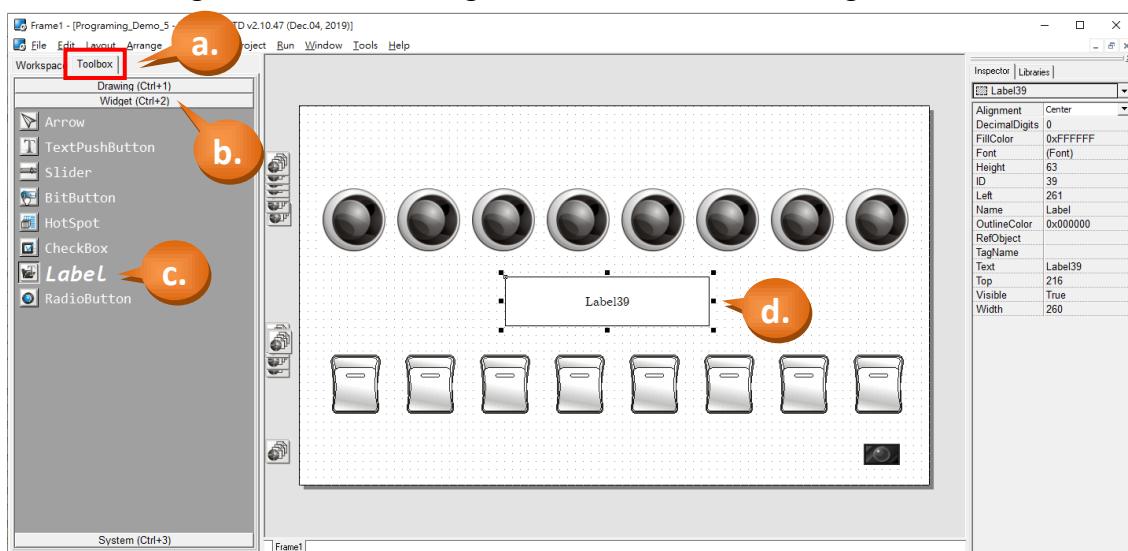
**Step 5:** Set up the communication detectors.

- As shown in the picture below, select the type of [graphic \(GUI\)](#) light from the “Libraries”.
- Select and drag the “Dev\_M\_7055\_1\_ERROR” item from the Workspace panel into the frame. (The light will turn on if there are any communication problems between the TouchPAD and the I/O model.)



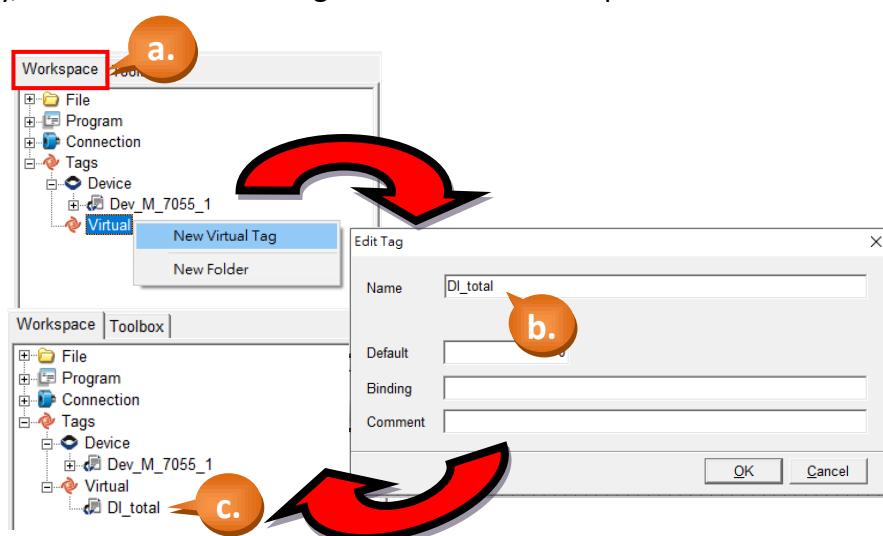
**Step 6: Create a Label to display the sum of all DI tag.**

- Switch to the “Toolbox” tab, as shown in the picture below.
- Click the “Widget (Ctrl+2)” heading to unfold the “Toolbox” panel.
- Select the “Label” icon, then move your mouse to the design area. You will notice that the cursor becomes a “+”.
- In the design area, click and drag the mouse to draw a rectangle.

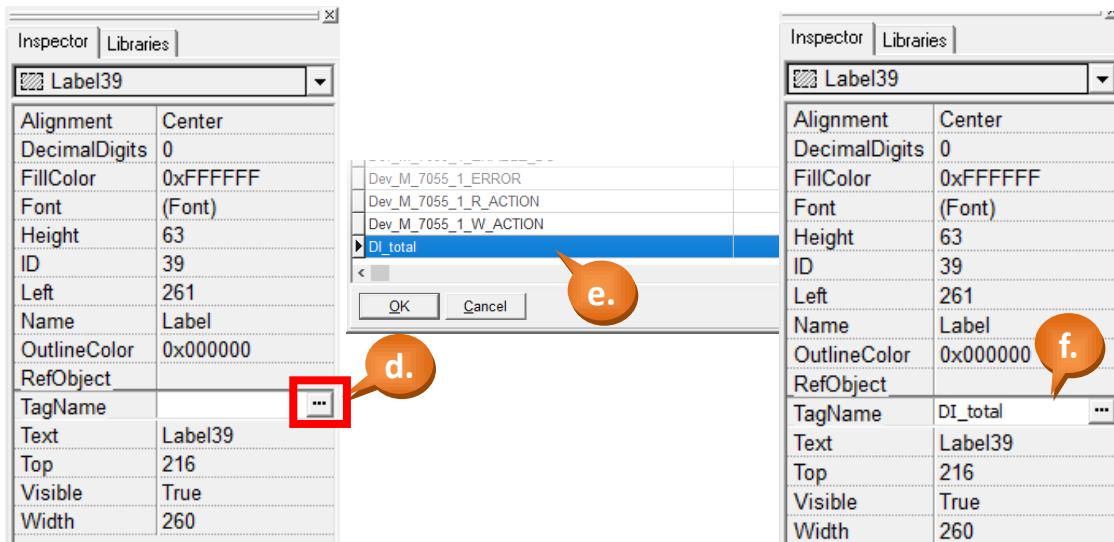


**Step 7: Create the corresponding Virtual Tag.**

- As shown in the picture below, right click the “Virtual” in the Workspace to create the tags.
- Enter the name of the tag in the “Edit variable” window. Here we use “DI\_total”.
- Finally, we can see that the tag created in the Workspace.

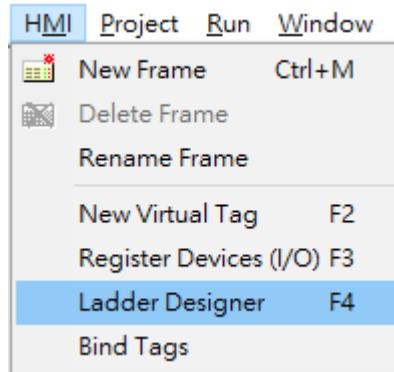


- d. Click the “Label39” and then click the “TagName” property in the Inspector and there shows a little button “...”. Click on the “...” button.
- e. Select the tag you want. Here we select the tag, “DI\_total”.
- f. Finally, we can see the tag we just choose



**Step 8: Use the Ladder Designer to program the logic.**

- a. Click the menu [HMI] > [Ladder Designer].



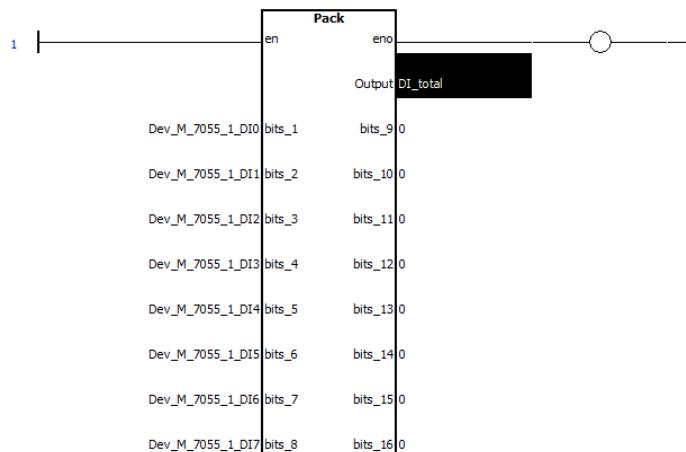
- b. Press the F7 button (or F7 key on the keyboard) to create a new function block.



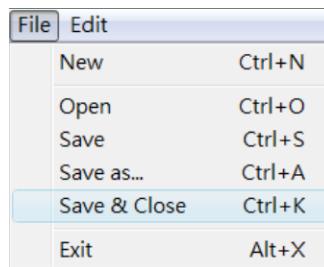
- c. Double-click on the function block to set the function to it. Here we set it the function, “Pack”, of the category, “user\_define”.

| Function Block     |                   |                   |                                                               |
|--------------------|-------------------|-------------------|---------------------------------------------------------------|
|                    | Function Name     | Display Name      | Caption                                                       |
| convert            | VOLTAGE           | VOLTAGE           | VOLTAGE                                                       |
| counter            | GotoFrame         | GotoFrame         | Go to Frame Number                                            |
| default            | CurrentFrame      | CurrentFrame      | Get Current Frame Number                                      |
| math               | 240x320_BLOff     | 240x320_BLOff     | 240x320_BLOff function                                        |
| system             | 240x320_BLOn      | 240x320_BLOn      | 240x320_BLOn function                                         |
| timer              | 320x240_BLOff     | 320x240_BLOff     | 320x240_BLOff function                                        |
| <b>user_define</b> | 320x240_BLOn      | 320x240_BLOn      | 320x240_BLOn function                                         |
| VPD-130            | SetTimeOut        | SetTimeOut        | SetTimeOut function                                           |
|                    | Timeout_To_Action | Timeout_To_Action | Timeout_To_Action function                                    |
|                    | WORD2Float        | WORD2Float        | WORD2Float                                                    |
|                    | Float2WORD        | Float2WORD        | Float2WORD                                                    |
|                    | WORD2DWORD        | W2DW              | WORD2DWORD                                                    |
|                    | DWORD2DWORD       | DW2W              | DWORD2DWORD                                                   |
|                    | ParamsWrite       | Params_Write      | ParamsWrite; Be Careful! Flash write limit only 100,000 times |
|                    | ParamsRead        | Params_Read       | ParamsRead                                                    |
|                    | <b>Pack</b>       | <b>Pack</b>       | <b>Pack</b>                                                   |
|                    | Unpack            | Unpack            | Unpack                                                        |

- d. Double-click on the neighborhood of the input parameters “bits\_1” to “bits\_16” and adding the corresponding Device Tag, as shown in the picture below.



Step 9: Save and Close (the Ladder).



Step 10: Build and Download the demo to the TouchPAD model.  
Please refer [2.4 Downloading Methods for TouchPAD](#)

### 3.3 Demo List

We have introduced you four basic demos in the previous chapter respectively. Please refer the demo list to view the more detailed application.

<http://www.icpdas.com/en/download/show.php?num=1000&nation=US&kind1=&model=&kw=HMIWORKS>

| Demo                                 | Descriptions                                                                                           |
|--------------------------------------|--------------------------------------------------------------------------------------------------------|
| <a href="#">Convert</a>              | This demo shows how to convert between data types                                                      |
| <a href="#">Demo All Widgets</a>     | This demo shows how to use widgets in HMIWorks.                                                        |
| <a href="#">Digital Clock</a>        | This demo makes TouchPAD a digital clock with a setting page for date and time.                        |
| <a href="#">Extern</a>               | How to use the variables of the other frames.                                                          |
| <a href="#">Geometry API</a>         | This demo shows how to use the geometry function on the paint box.                                     |
| <a href="#">ICPDAS Device</a>        | How to control the ICPDAS products, e.g., ET-7000, I-7000, M-7000, LC, PM and tM Series.               |
| <a href="#">Ladder tags</a>          | How to read/write the Ladder tags.                                                                     |
| <a href="#">Modbus</a>               | This demo makes TouchPAD as a Modbus TCP/RTU Master/Slave device in the C programming language.        |
| <a href="#">Other</a>                | The more project example.                                                                              |
| <a href="#">PassWord</a>             | Create a virtual password button, the user needs to enter the correct password to continue next steps. |
| <a href="#">Rotary Switch</a>        | How to read the value of the rotary switch.                                                            |
| <a href="#">ScreenSaver</a>          | This demo shows how to run a screen saver and then how to go back to the normal mode program.          |
| <a href="#">Set IP From Flash</a>    | Get/Set the network configuration of the TouchPAD from the parameter area in the internal flash.       |
| <a href="#">UART</a>                 | This demo test com1 and com2 send/ receive function.                                                   |
| <a href="#">User Flash</a>           | This demo shows how to read/write data from/to the flashes.                                            |
| <a href="#">UserParams</a>           | Set/Get data from the 256-byte parameter area in the MCU (MicroController Unit) internal flash.        |
| <a href="#">Virtual Keyboard</a>     | Create a virtual keyboard.                                                                             |
| <a href="#">VPD Panelkey XVboard</a> | How to use the Panelkey and XV-board, the demo only support the VPD device.                            |
| <a href="#">ebFonts</a>              | <a href="#">How to transmit or show multilingual text with ebFont on TouchPAD.</a>                     |
| <a href="#">TCP/IP</a>               | How to use the TouchPAD as TCP Client/Server.                                                          |
| <a href="#">MQTT</a>                 | This demo shows how to use the MQTT API.                                                               |

# Appendix A. Downloading Information

This manual is only a brief introduction to the TouchPAD functions. Please visit the following websites for more detailed information about the TouchPAD series.

User Manual:

[http://www.icpdas.com/web/product/download/software/development\\_tool/hmiworks/document/manual/HMIWorks\\_Software\\_User\\_Manual\\_en.pdf](http://www.icpdas.com/web/product/download/software/development_tool/hmiworks/document/manual/HMIWorks_Software_User_Manual_en.pdf)

Products Website & Datasheets:

[http://www.icpdas.com/en/product/guide+Panel\\_Products+TouchPAD+TPD\\_Series](http://www.icpdas.com/en/product/guide+Panel_Products+TouchPAD+TPD_Series)

FAQ:

<http://www.icpdas.com/en/faq/index.php?type=&model=&fkw=HMIWORKS#882>

I-7000 I/O Modules (Support for the DCON Protocol) :

[http://www.icpdas.com/en/product/guide+Remote\\_I\\_O\\_Module\\_and\\_Unit+RS-485\\_I\\_O\\_Modules+I-7000](http://www.icpdas.com/en/product/guide+Remote_I_O_Module_and_Unit+RS-485_I_O_Modules+I-7000)

M-7000 I/O Modules (Support for the Modbus RTU & DCON Protocols) :

[http://www.icpdas.com/en/product/guide+Remote\\_I\\_O\\_Module\\_and\\_Unit+RS-485\\_I\\_O\\_Modules+I-7000](http://www.icpdas.com/en/product/guide+Remote_I_O_Module_and_Unit+RS-485_I_O_Modules+I-7000)

# Appendix B. Creating Your Own GUI

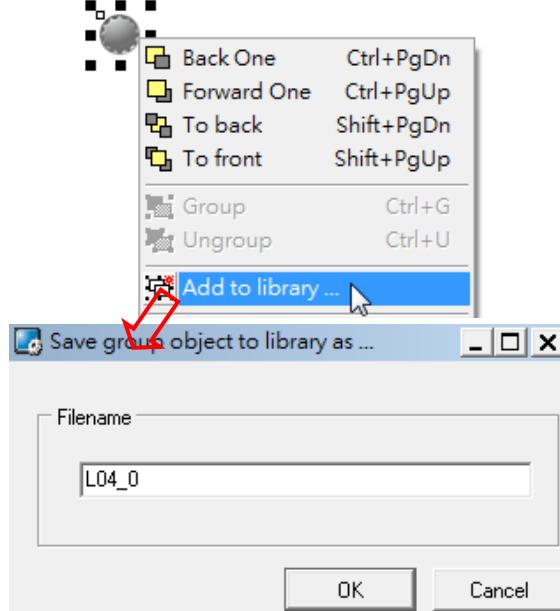
The HMIWorks Libraries provide many images and icons for integration into the GUI (Graphical User Interface), and users can also create their own custom GUIs.

- a. First, open an image file in MS “Paint”, then click the toolbox to select the entire image (or press Ctrl+A on the keyboard) and then choose the copy option from the “Edit” menu (or press Ctrl+C on the keyboard) to copy the picture.



- b. Select the Paste option from the HMIWorks “Edit” menu (or press Ctrl+V on the keyboard) to paste the image to the HMI frame, then right click on the image and select “Add to library...” from the popup menu. Enter a name, such as “L04\_0” (In this example, the light has two states: 0 means OFF, 1 means ON), as per the examples below.

- c. Follow the same procedure to create a second image for the ON state (L04\_1).



# Revision History

| Revision | Date      | Description                                                                                                                                                                                                                                                                                                                                                                                      |
|----------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0.5    | Apr. 2012 | Initial issue                                                                                                                                                                                                                                                                                                                                                                                    |
| 1.0.6    | Jun. 2020 | <ul style="list-style-type: none"><li>➤ Update the screenshot of HMIWorks(v2.10.46)</li><li>➤ Modify the content to the universal version.</li><li>➤ Divide the demos of the C language and Ladder to two chapters.</li><li>➤ Add and remove some demos.</li><li>➤ Remove the content of M3 model and add the content of M4 model.</li><li>➤ Update all links of the official website.</li></ul> |