Chapter 6   Cash Register with HMI

Design a simple cash register similar to one found at McDonald's or Burger King. To do this, determine a menu of five or six items from the restaurant. Also, include a Total button or a clear button or possibly both. Also, include a means for backing out of a mistake without starting over from zero. Display the cost of the total order in the PLC at an address in the data table. Use Floating Point Math with two decimal places.

For example:

Whopper Combo	Whopper	Cancel Last
Whopper Dbl Combo	Fries	New Order
Whopper Jr Combo	Drink	Total/Tax/Op tional

Find the approximate prices from a McDonald's or Burger King for the items you choose. When an item is entered, its count is incremented automatically by one. If a button is entered multiple times, the count is incremented to display the total count. If a mistake is made, the attendant must be able to back up at least one entry and erase the last item or decrement that item by one.
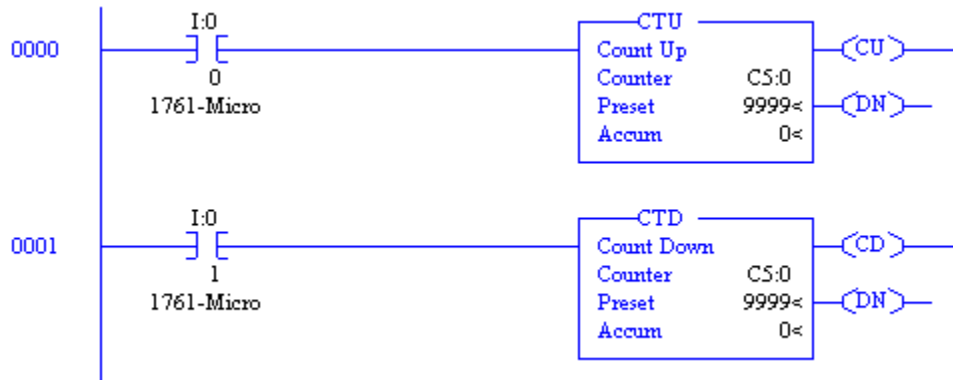
Display the final total in the PLC (not on the display of the trainer).

Options to the lab:

A Add logic for "To Go" order so that 6.25% tax is added if not "To Go".

B Add lights to buttons so that when an entry is made, the light lights.

C Add logic to keep track of total number of each entry for the day.

D Calculate profit for the day using your own profit numbers for each entry.

E Automatically recognize that the entry of the individual items such as Whopper, Fries, and Drink will be given the price of the Whopper Combo instead of the individual prices.
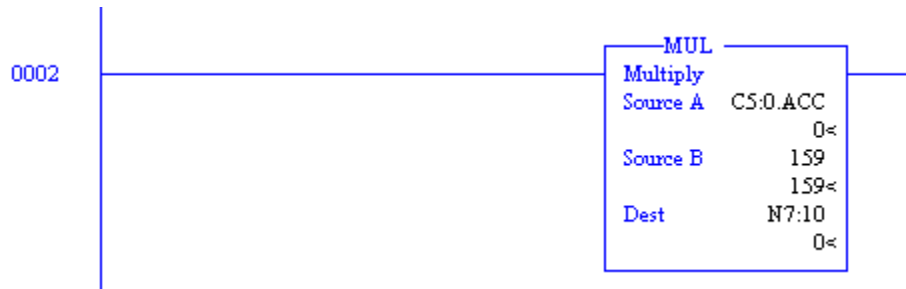
Hints to the base lab:

Notice that counters may be referenced as either Count Up or Count Down. If the count is counting up, the count is incremented in rung 0000. If the count is counted down, the count is decremented in rung 0001. Individual inputs are used to increment each product choice. However, to decrement the count, a separate button labeled "Cancel Last" is used. This button must remember the last product chosen and decrement that item. Use the logic in chapter 6 "Relay Instructions" to remember when a button was pushed.

```
              I:0                    ┌─CTU─────────────────┐
0000         ─┤ ├─                   │ Count Up            │──(CU)──
               0                     │ Counter      C5:0   │
            1761-Micro               │ Preset      9999<   │──(DN)──
                                     │ Accum          0<   │
                                     └─────────────────────┘

              I:0                    ┌─CTD─────────────────┐
0001         ─┤ ├─                   │ Count Down          │──(CD)──
               1                     │ Counter      C5:0   │
            1761-Micro               │ Preset      9999<   │──(DN)──
                                     │ Accum          0<   │
                                     └─────────────────────┘
```

The circuit above is for trial purposes only. Do not use it "as is" in the logic of programs.

The amount of each product is held in the counter Acc value. To access these values, use the addressing of C5:0.ACC (or C5.0.2).

Values of each product are multiplied by the amount of the item and the final total is summed together.

```
                                     ┌─MUL─────────────────┐
0002                                 │ Multiply            │
                                     │ Source A   C5:0.ACC │
                                     │               0<    │
                                     │ Source B      159   │
                                     │              159<   │
                                     │ Dest        N7:10   │
                                     │               0<    │
                                     └─────────────────────┘
```

The number in Source B may be either a constant (as is here), or a value from a N7 location. If from an N7 location, the value that is to be used must be entered into that N7 location.

You may use the Siemens' HMI instead of the wired buttons. More information about configuring the HMI may be found at:

http://www.youtube.com/watch?v=Gh0s4TIDGEE

Siemens SIMATIC S7-1200 Part 3 - Adding an HMI to a controller project: See how easy it is to integrate HMI screens into the controller user program using the same Step 7 Basic Software for both SIMATIC Basic HMI panels and S7-1200 Controllers. This is part three of a four part series.
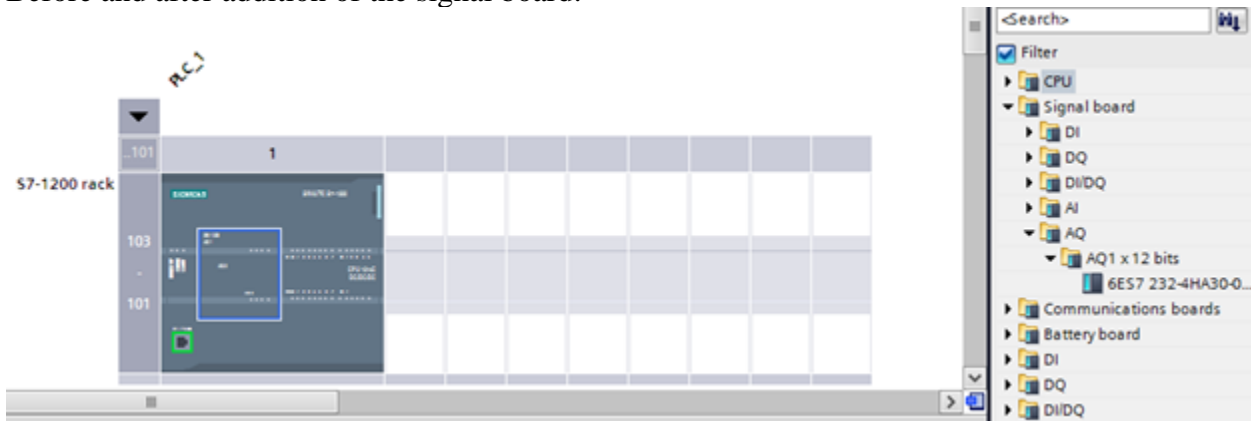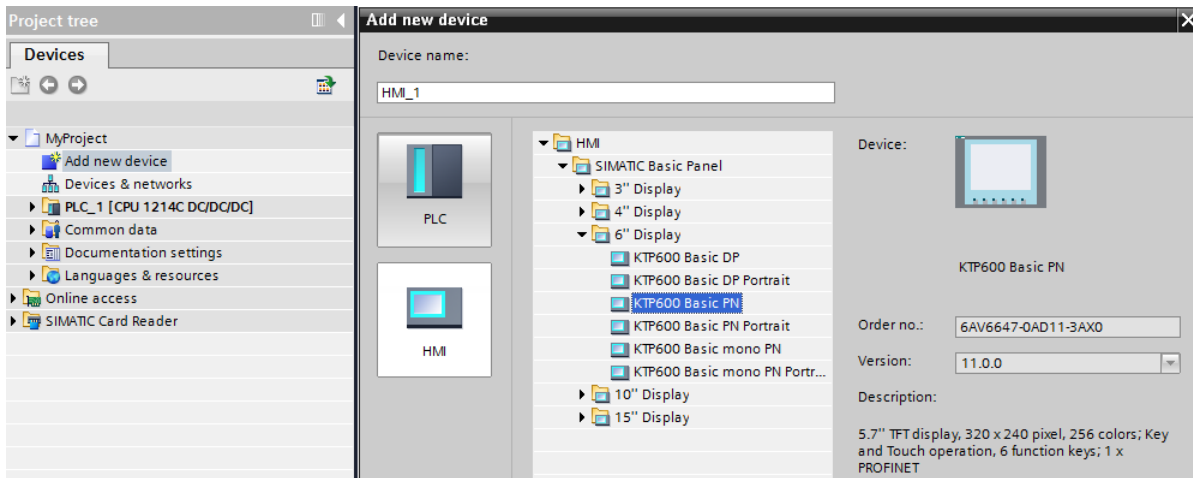
A Simple HMI Tutorial (Siemens)
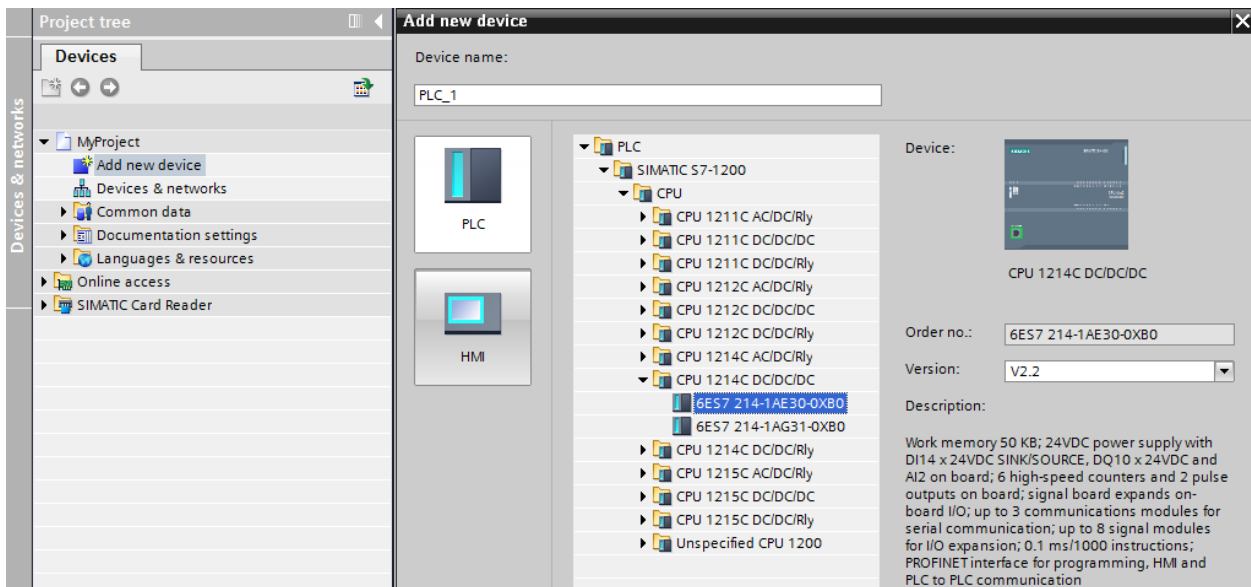
Again, the Siemens processor is to be configured similar to before:
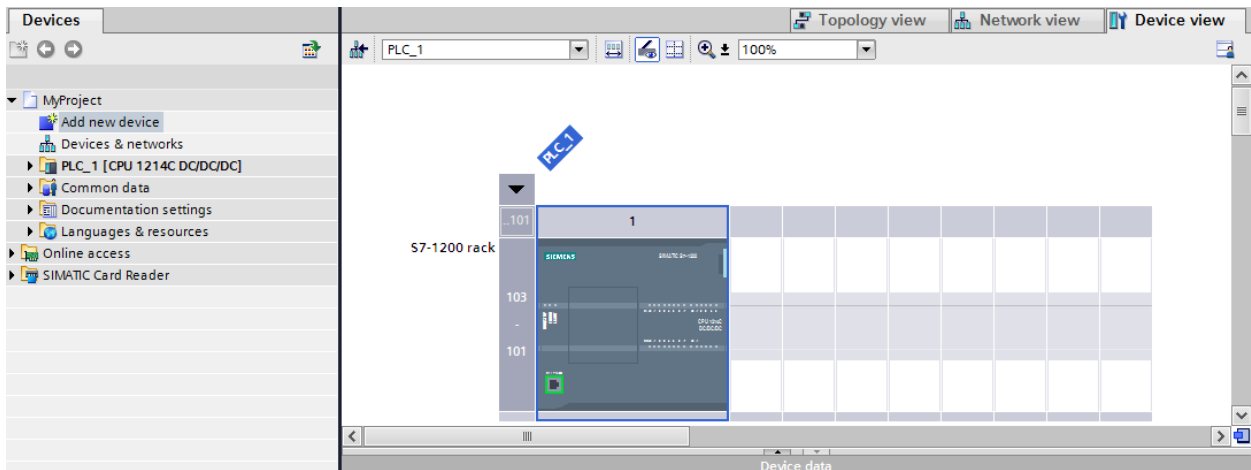
Before and after addition of the signal board:

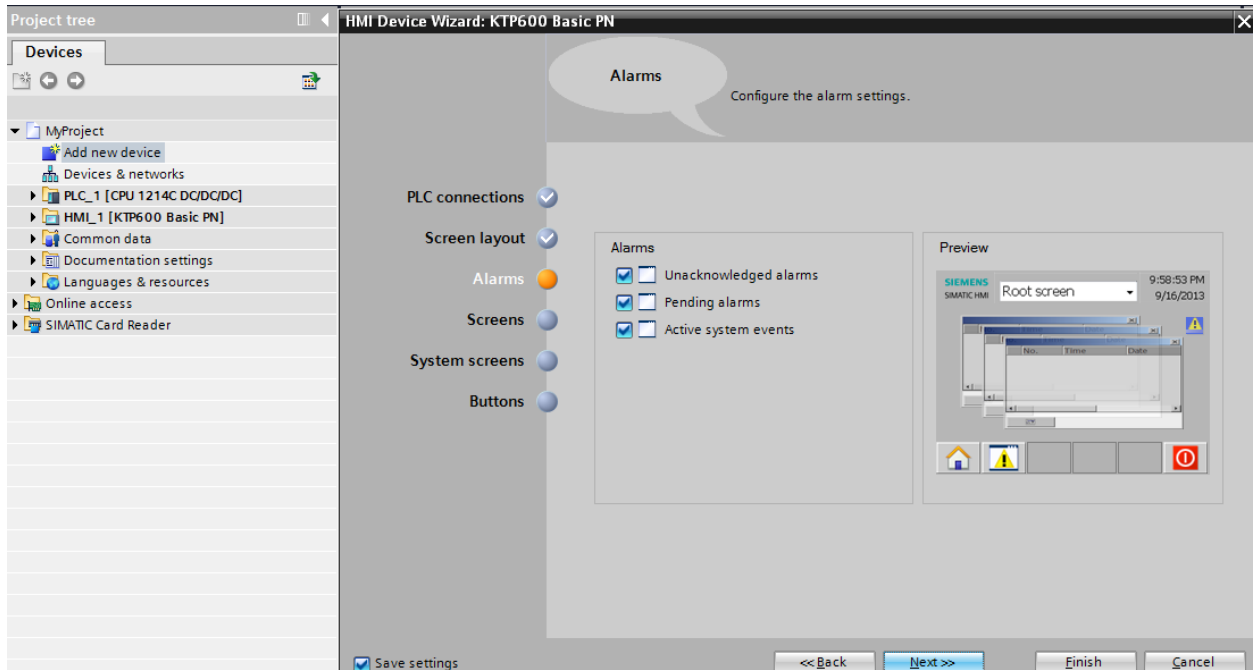Addition of the HMI used in the labs:

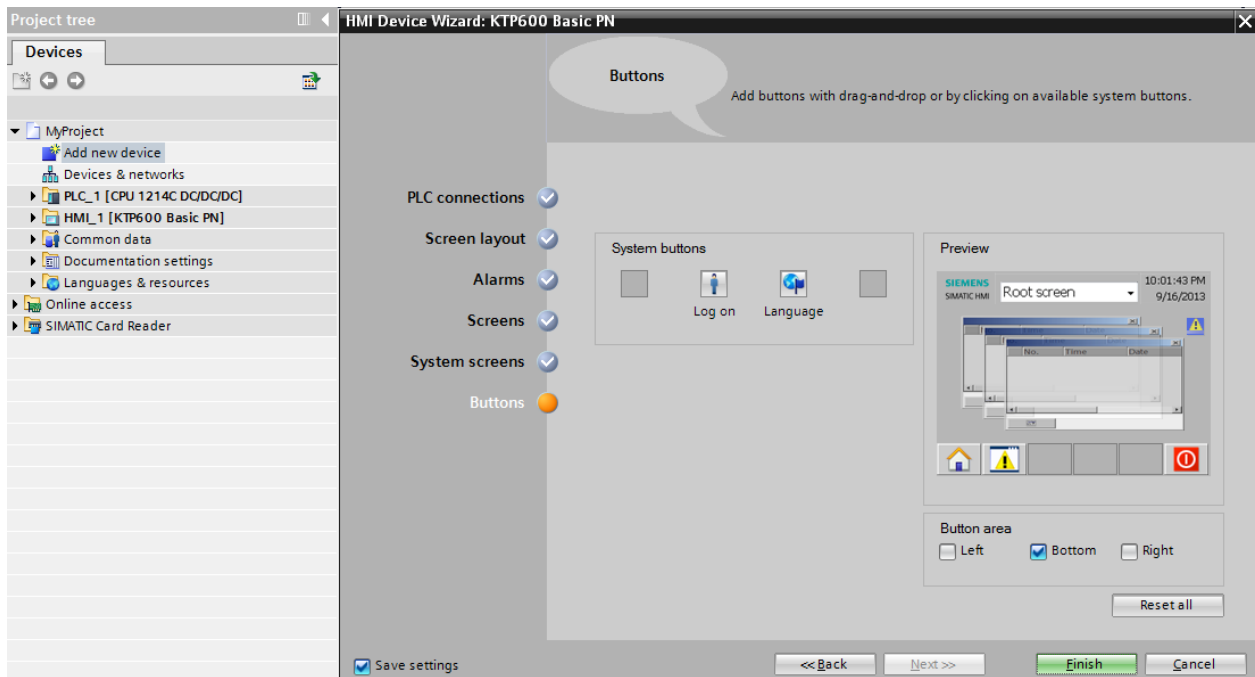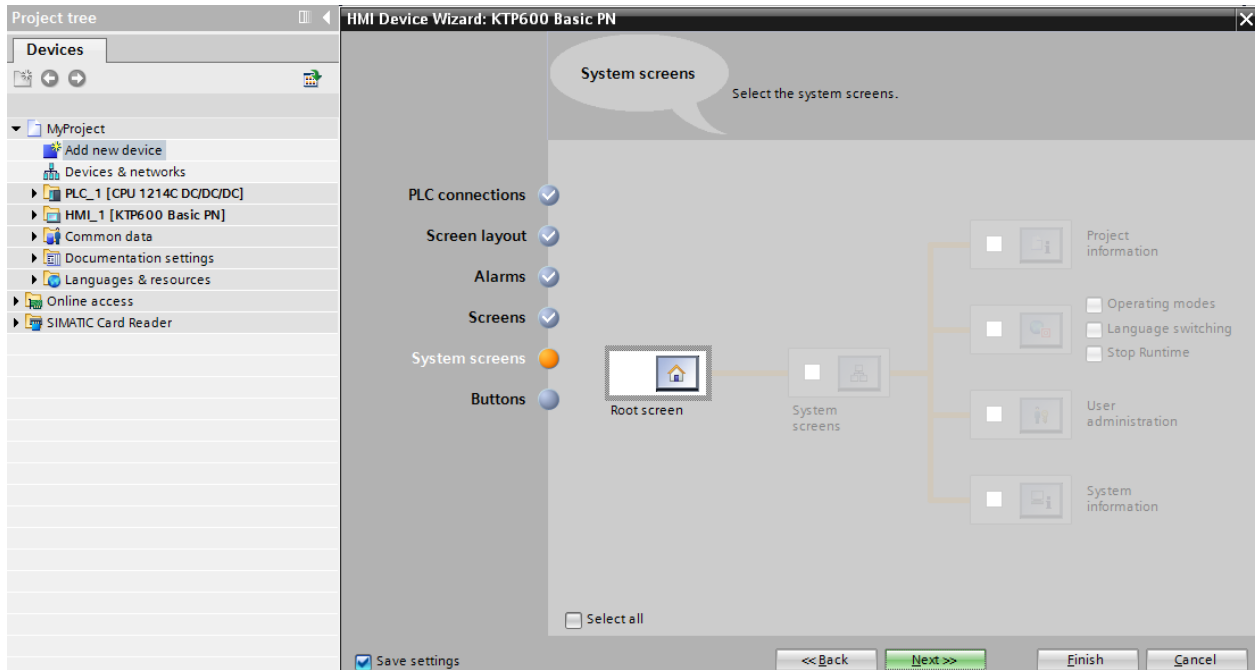The HMI Device Wizard:
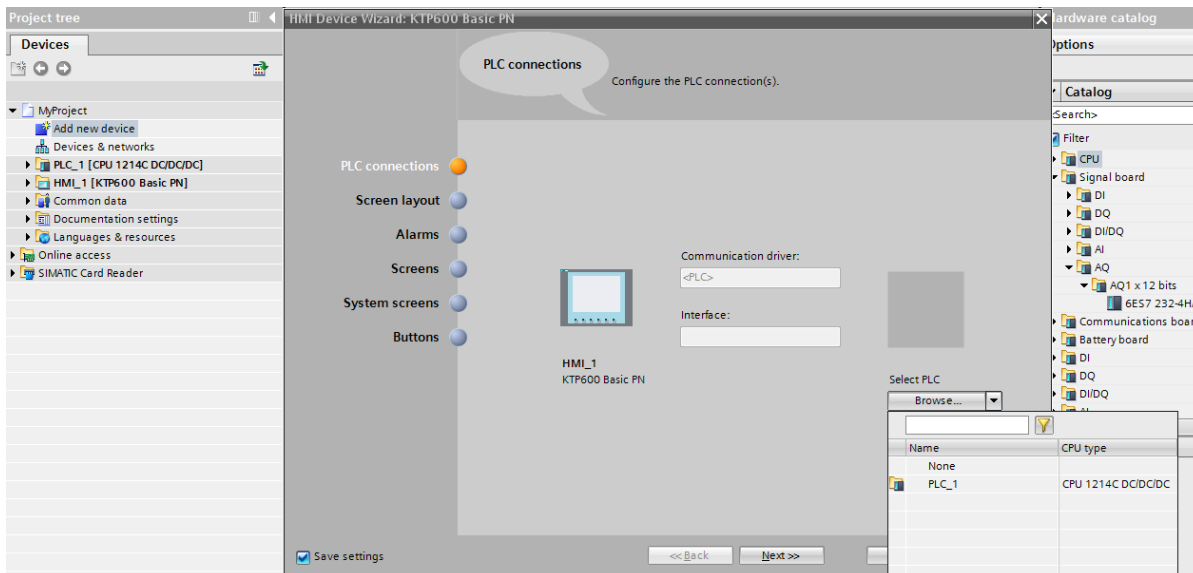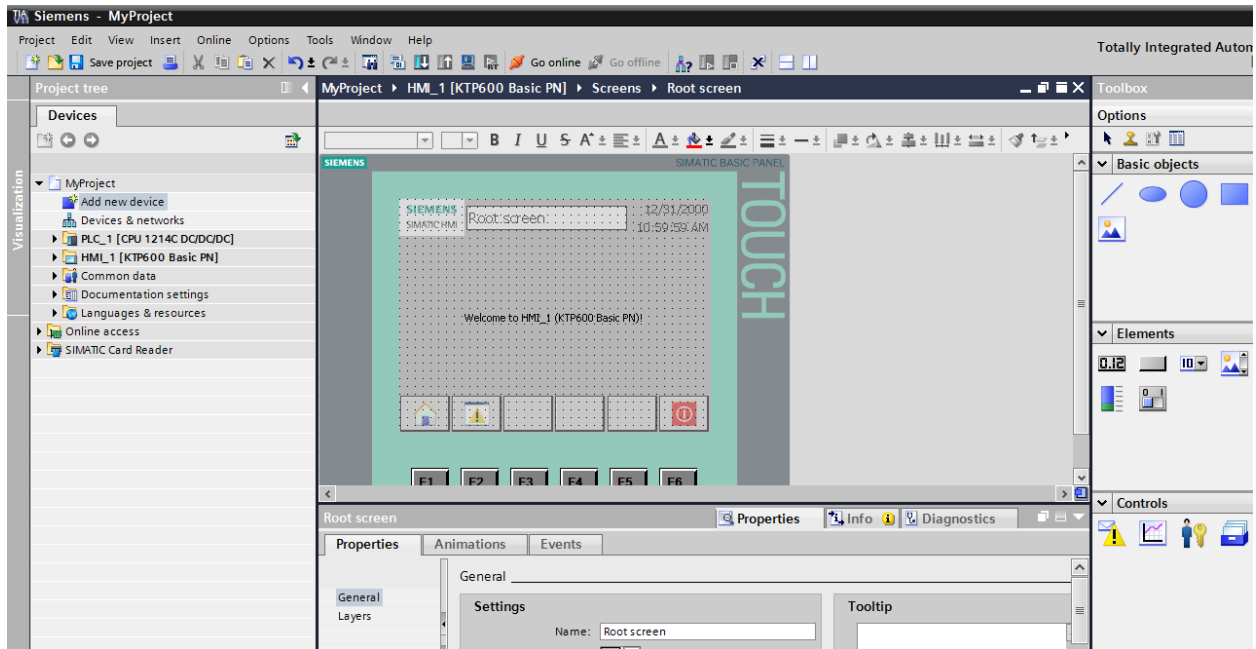
Keep answering Next>>
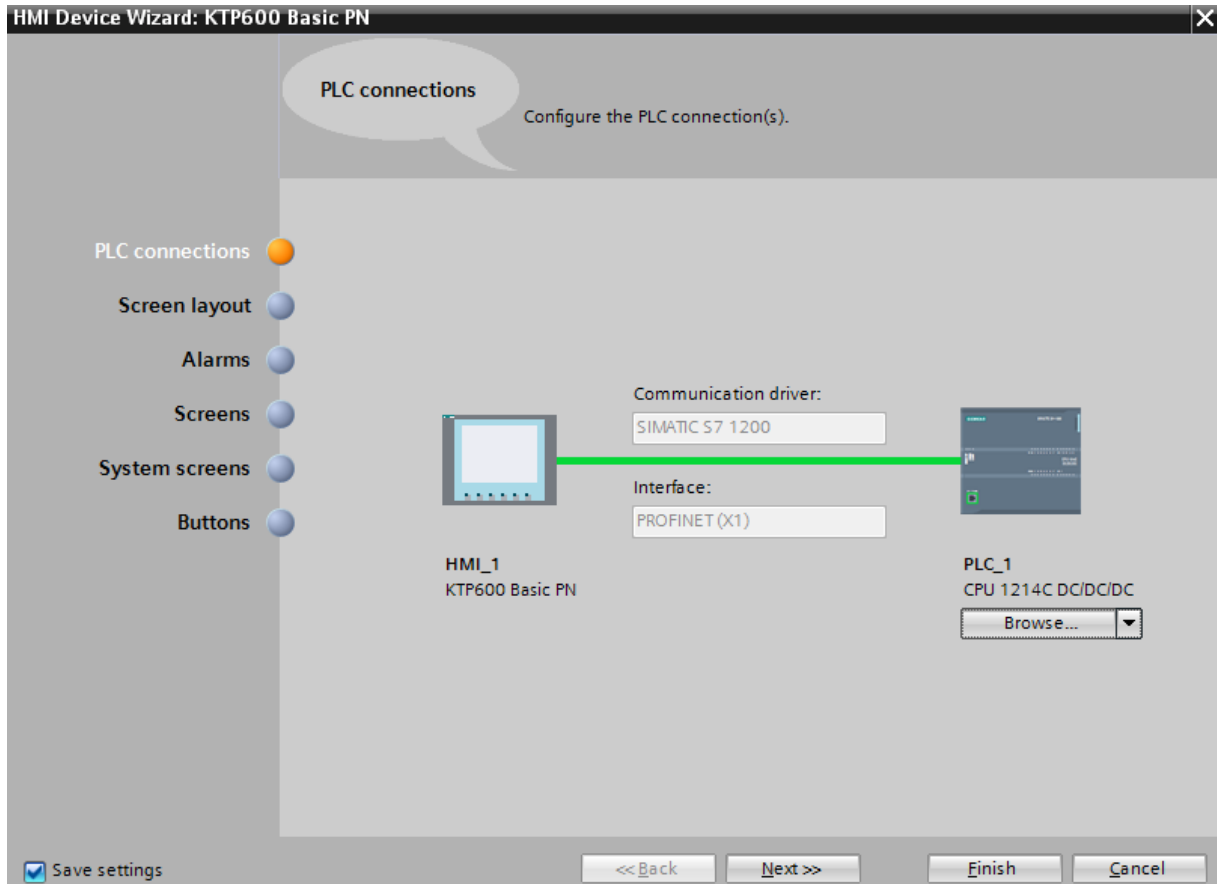
Keep answering Next>>
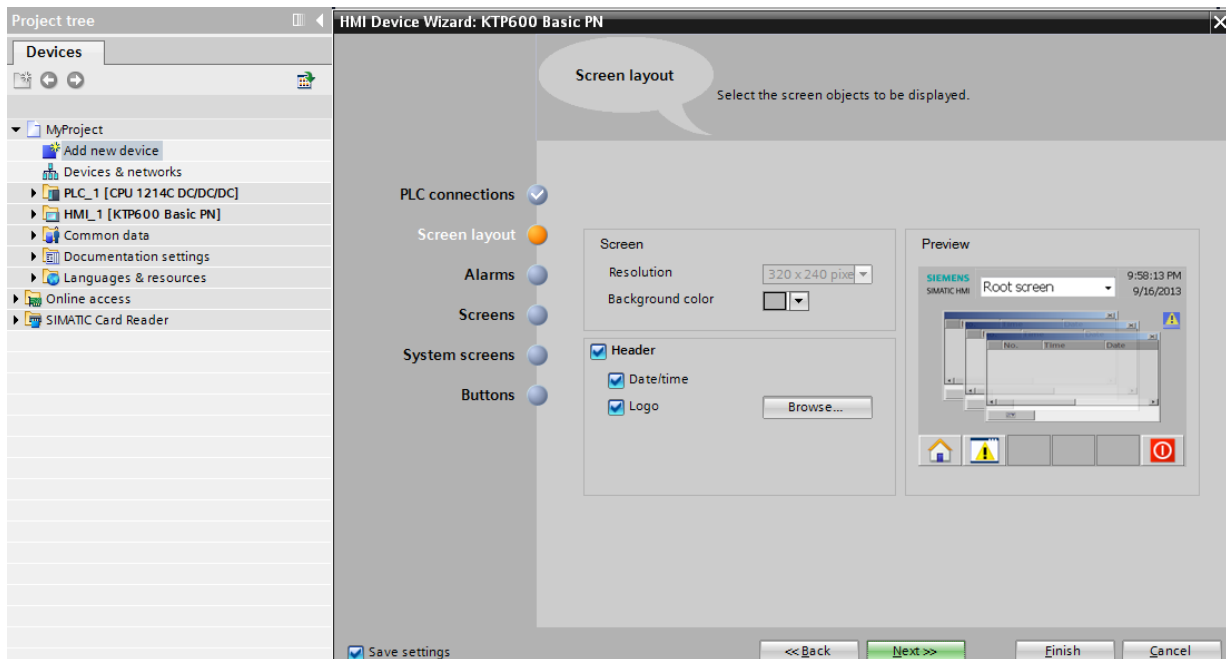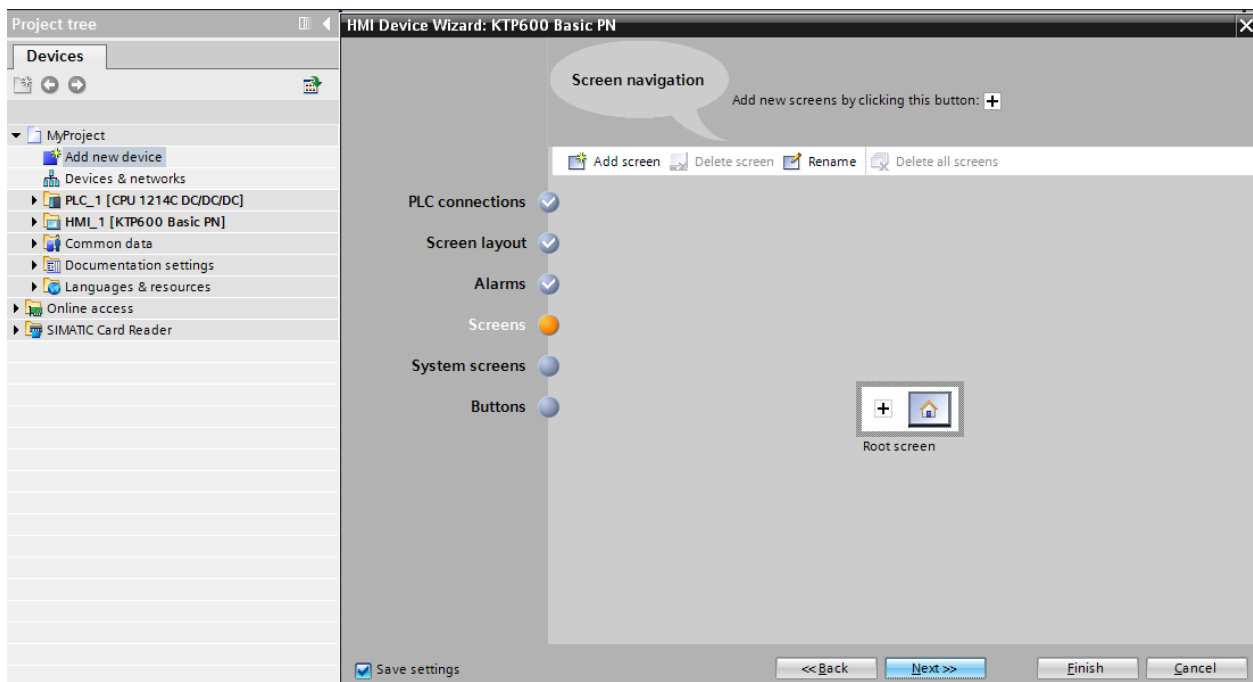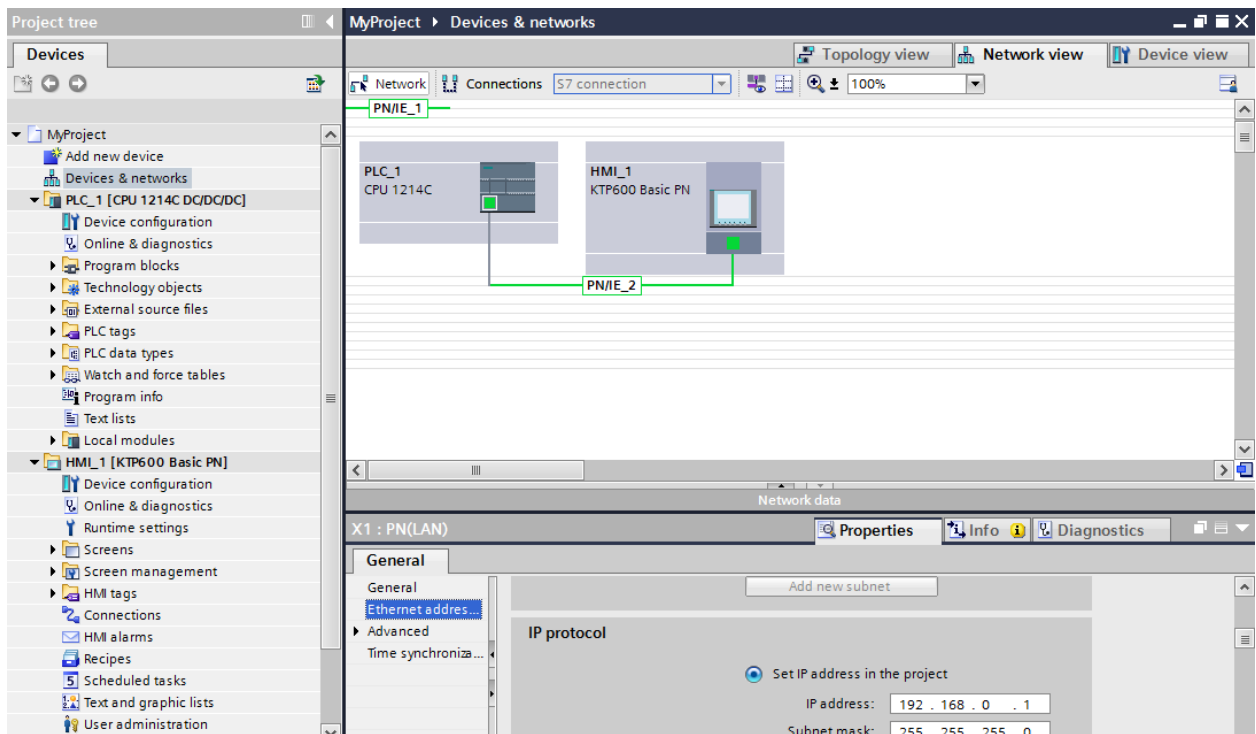
Keep answering Next>>

Then Finish

From the Devices and Networks choice in the Project Tree:

Choose Devices & networks

and set up both the IP address and Subnet mask for the PLC as well as the HMI. You may need to initialize the IP address of the HMI by setting the IP address up at power-up of the device. You have about 1 second to tap on the screen when power is first applied to get to the set-up screen. Set up the IP address of the HMI to 5 (192.168.0.5, 255.255.255.0). Read at the end of Ch.

15 about setting up a simulated HMI panel simulated on the computer screen.
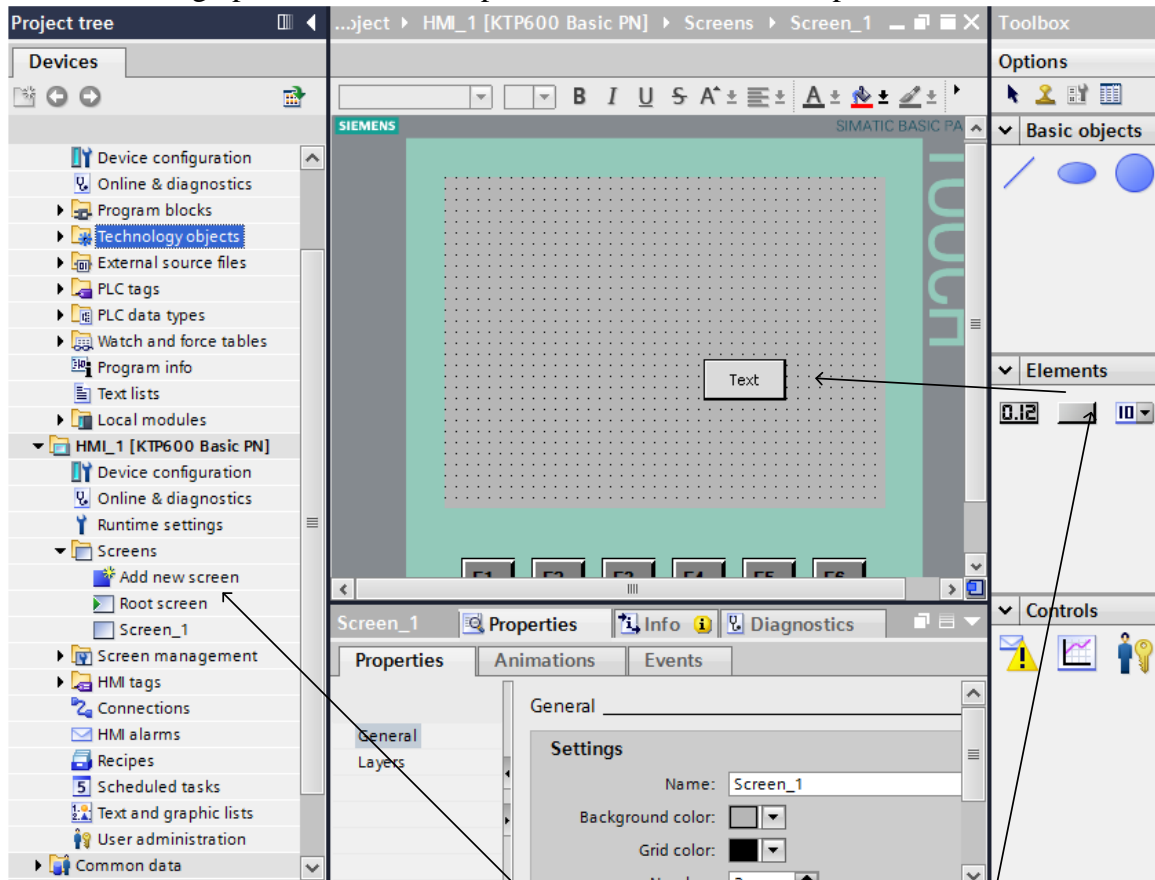
To add a new screen, double click on "Add new screen" in the Project Tree.
To begin a design, select a button from the Elements Toolbox at right. Drag the button onto the screen.

Entering a Button on the Screen and configuring the button to turn on a bit in the PLC

Button

The Button object allows you to configure an object that the operator can use in runtime to execute any configurable function.

Button Layout

In the Inspector window, you customize the position, geometry, style, color and font types of the object. You can adapt the following properties in particular:

- Mode: Defines the graphic representation of the object.
- Text / Graphic: Defines whether the Graphic view is static or dynamic.
- Define hotkey: Defines a key, or shortcut that the operator can use to actuate the button.

You can only define a hotkey for HMI devices with keys.

Mode for Button

The button display is defined in Properties > Properties > General >Mode in the Inspector window.

Mode Description

Invisible   The button is not visible in runtime.
Text   The button is displayed with text. This text explains the function of the
                 button.
Graphic   The button is displayed with a graphic. This graphics represents the
                 function of the button.

Depending on the device, Text /Graphic is also available.

The Mode property settings are used to define whether the display is static or dynamic. The
display is defined in Properties > Properties > General >Text or Graphic in the Inspector window.
Your options for the type Graphic include the following.

Type	Option	Description	
Graphic	Graphic	Graphic OFF	is used to specify a graphic that is displayed in the button when the state is "OFF". If you enable Graphic ON, you can enter a graphic for the ON state.
		Graphics list	The graphic in the button depends on the state. The entry from the graphics list corresponding to the state is displayed.

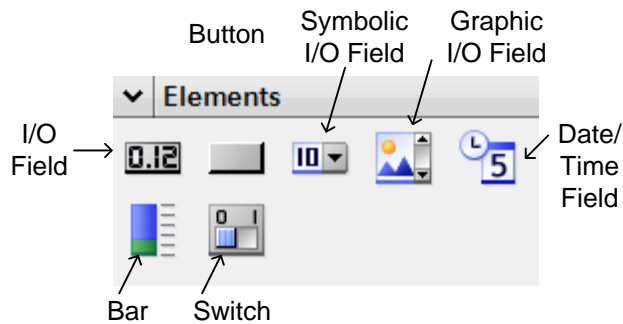To turn on the bit in the PLC, use Press:
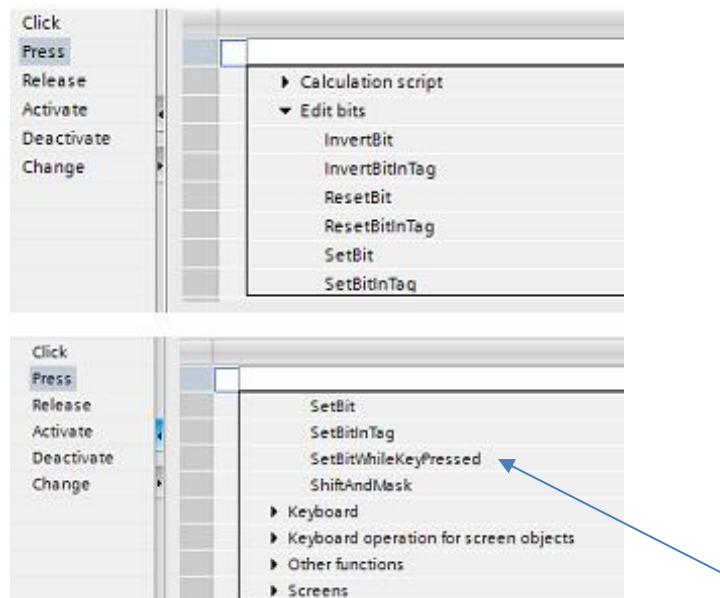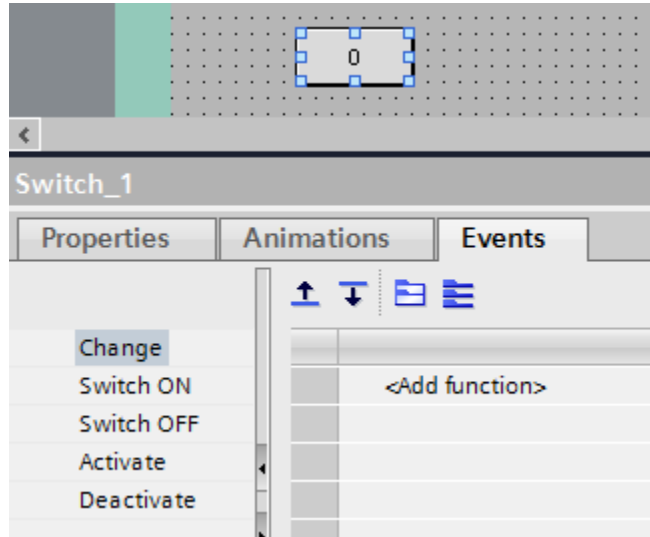
and then SetBitWhileKeyPressed:
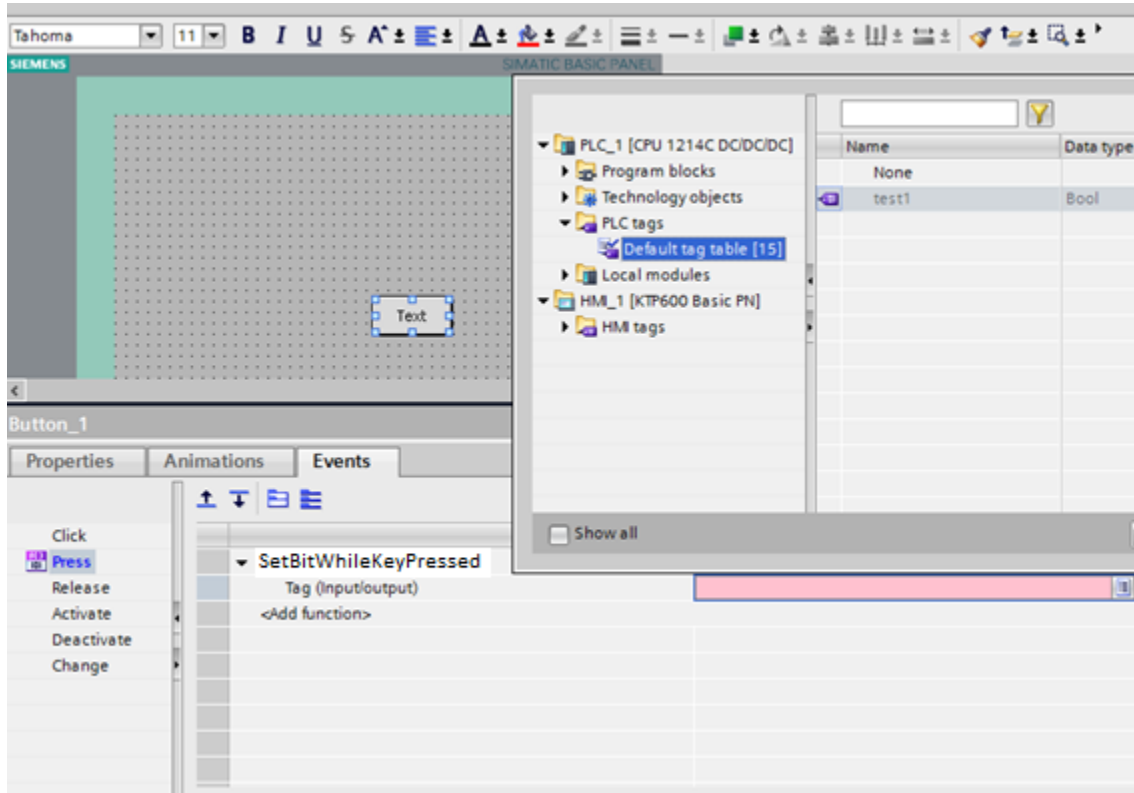
The tag is built in the PLC for an internal bit and referenced to the SetBitWhileKeyPressed function:

There are a number of input types for data entry from the HMI. They include:

For Switch, the following choices are available:

Screen Navigation

You will also need to consider configuring screen navigation. For a production process consisting of multiple sub-processes, you will configure multiple screens. You have the following options to enable the operator to switch from one screen to the next in Runtime:

- Assign buttons to screen changes
- Configuring screen changes at local function keys

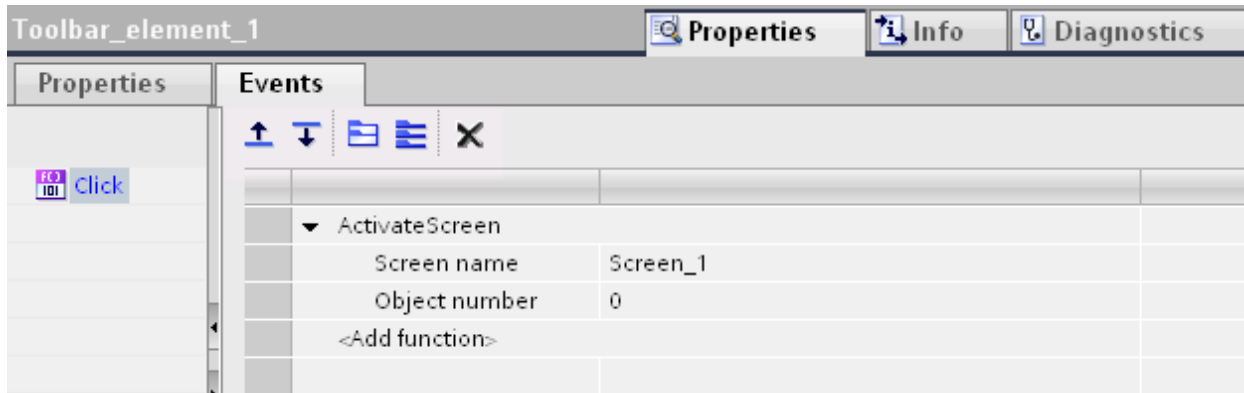The procedure for configuring screens follows:

Before you create a screen change, define the plant structure and derive from it the screen changes that you want to configure.

Create the start screen under Runtime Settings > General > Start screen.

You will need to assign a button to change the screen. You will need to configure a button in the screen to switch between the screens on the HMI device during operation.

Procedure

  1. Double-click Screen_1 in the project navigation.
  2. Move Screen_2 from the project tree to the open screen by drag&drop. A button with the name Screen_1 is inserted.
  3. In the Inspector window, select Properties > Events > Click. The ActivateScreen system function is displayed in the "Function list".

Toolbar_element_1				Properties	Info	Diagnostics
Properties	Events					
	↑ ↓ ⊟ ≣ ✕					
Click		▼ ActivateScreen				
			Screen name	Screen_1		
			Object number	0		
		<Add function>				

  4. At the Object number attribute, define, if required, the tab sequence number of the object on which the focus is to be set after a screen change. You can also specify a tag that contains the object number.

Overview of HMI tag tables

HMI tag tables contain the definitions of the HMI tags that apply across all devices. A tag table is created automatically for each HMI device created in the project. In the project tree there is an HMI tags folder for each HMI device. The following tables can be contained in this folder:

  ● Standard tag table
  ● User-defined tag tables
  ● All tags

The following tables are also available in an HMI tag table:

  ● Discrete alarms
  ● Analog alarms

With the help of these tables you configure alarms for the currently selected HMI tag.
In the project tree you can create additional tag tables in the HMI tags folder and use these to sort and group tags and constants. You can move tags to a different tag table using a drag&drop operation or with the help of the Tag table field. You activate the Tag table field using the shortcut menu of the column headings.
Standard tag table

There is one standard tag table for each HMI device of the project. It cannot be deleted, renamed or moved. The standard tag table contains HMI tags and, depending on the HMI device, also system tags. You can declare all HMI tags in the standard tag table, or create additional user-defined tag tables as you want.

User-defined tag tables

You can create multiple user-defined tag tables for each HMI device in order to group tags according to your requirements. You can rename, gather into groups, or delete user-defined tag tables. To group tag tables, create additional subfolders in the HMI tags folder.

All tags

The All tags table shows an overview of all HMI tags and system tags of the HMI device in question. This table cannot be deleted, renamed or moved.

Discrete alarms table

In the Discrete alarms table, you configure discrete alarms to the HMI tag selected in the HMI tag table. When you configure a discrete alarm, multiple selections in the HMI tag table is not possible. You configure the discrete alarms for each HMI tag separately.
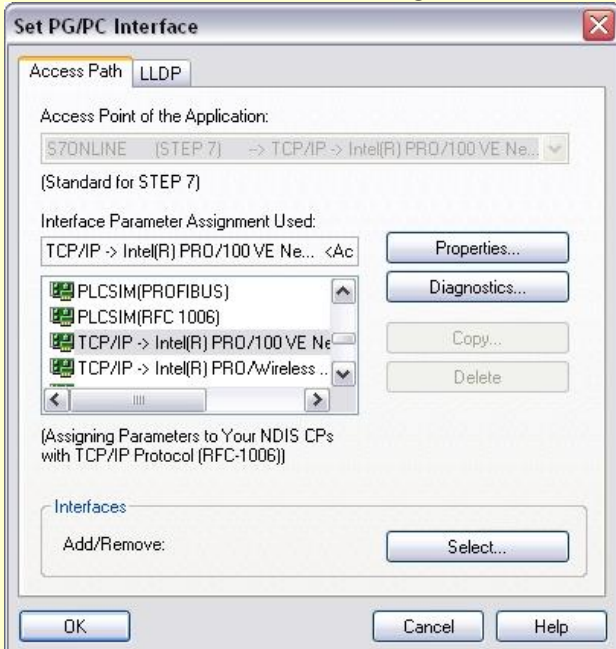
Analog alarms table

In the Analog alarms table, you configure analog alarms to the HMI tag selected in the HMI tag table. When you configure an analog alarm, multiple selections in the HMI tag table is not possible. You configure the analog alarms for each HMI tag separately.

Defining Limits for a Tag

For numerical tags, you can specify a value range by defining a low and high limit. Additionally, you configure the system to process a function list whenever a tag value drops below or exceeds its configured value range.

Setting the PG/PC Interface for Runtime Simulation

In order to establish a connection between runtime simulation at the PG/PC and theS7-1200 CPU, first we have to set the PG/PC interface to TCP/IP.

No.	How it's done
1	Open the system control • with "Start > System control" (start menu for the simplified access to the programs under Windows XP) • or with "Start > Settings > System control" (for the classical start menu as in earlier Windows versions).
2	Now double click on the icon "Set PG/PC interface"
3	In the tab "Access Path", set the following parameters: 1. For the access point of the application, select from the drop down menu "S7ONLINE [STEP 7]". 2. In the list of Interface Parameter Assignment Used, highlight the interface "TCP/IP(Auto) -> with your network card that is connected directly to the panel and the control system; for example, Intel(R) PRO/100 VE. 3. Then click OK and confirm the message that follows with OK also.

Starting the Configuration in Runtime

Click on the button Start runtime.

Alternately, the following must be set up to get the PID program to run properly with the HMI simulate mode:
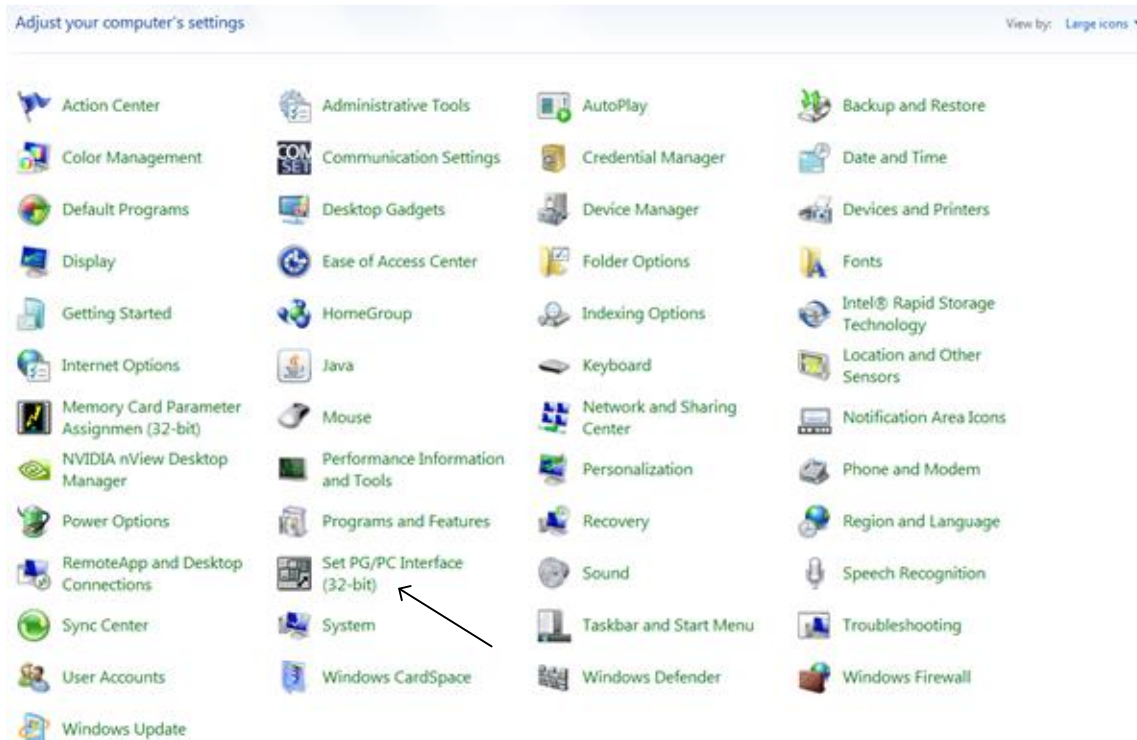

Click on the SetPG/PC Interface box above:
We are going to use the Siemens program TIA Portal V14 in order to run the program given for the ball-in-tube lab. This program will be used to download the PLC program but not the HMI program.

Choose the third of the Broadcom choices. Click OK.
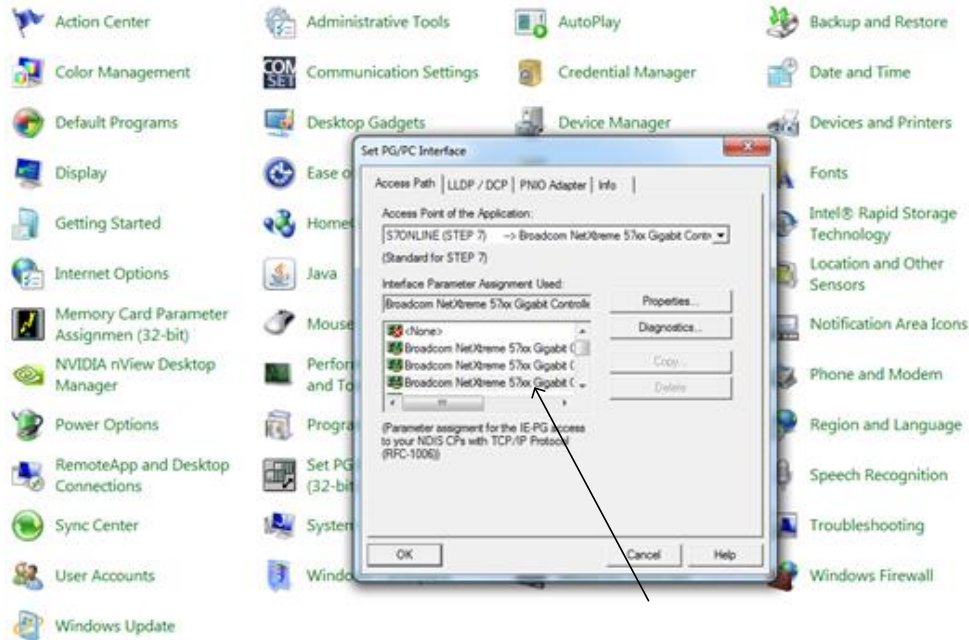
This allows the Siemens program to run the HMI program in simulate mode. Then download the program to the PLC. Do not download the HMI program since we do not have the HMI to download to.
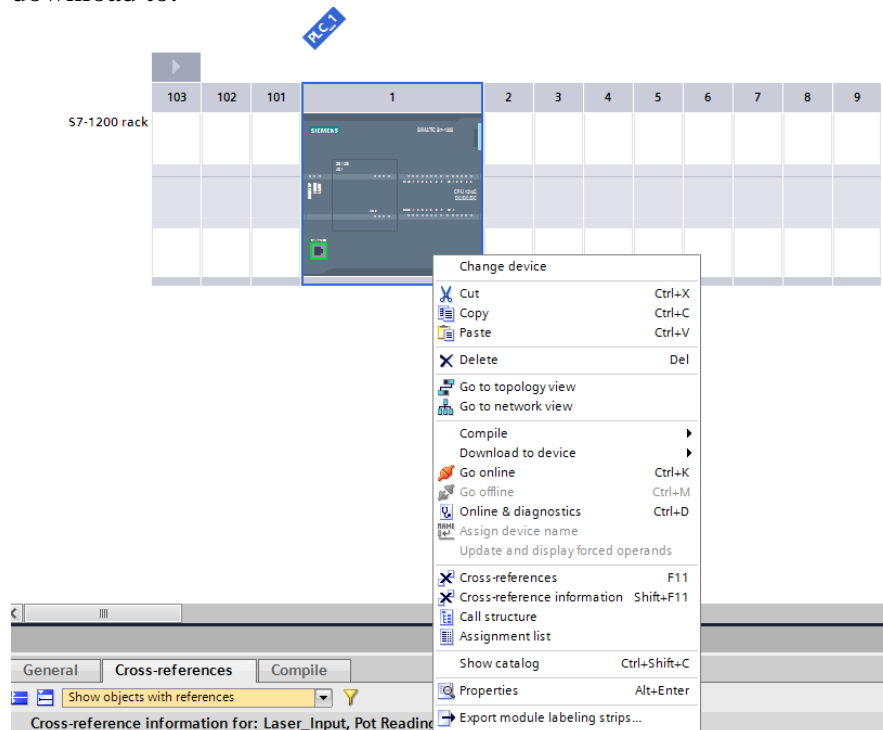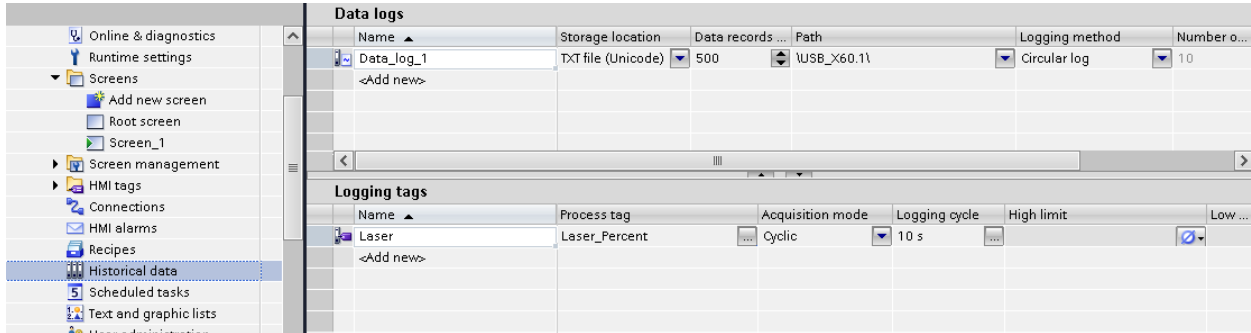

Then click on the HMI's Screen, Screen_1. Notice the Start Simulation button turn blue. It now allows the student to run the HMI via simulation mode from the screen of the pc.
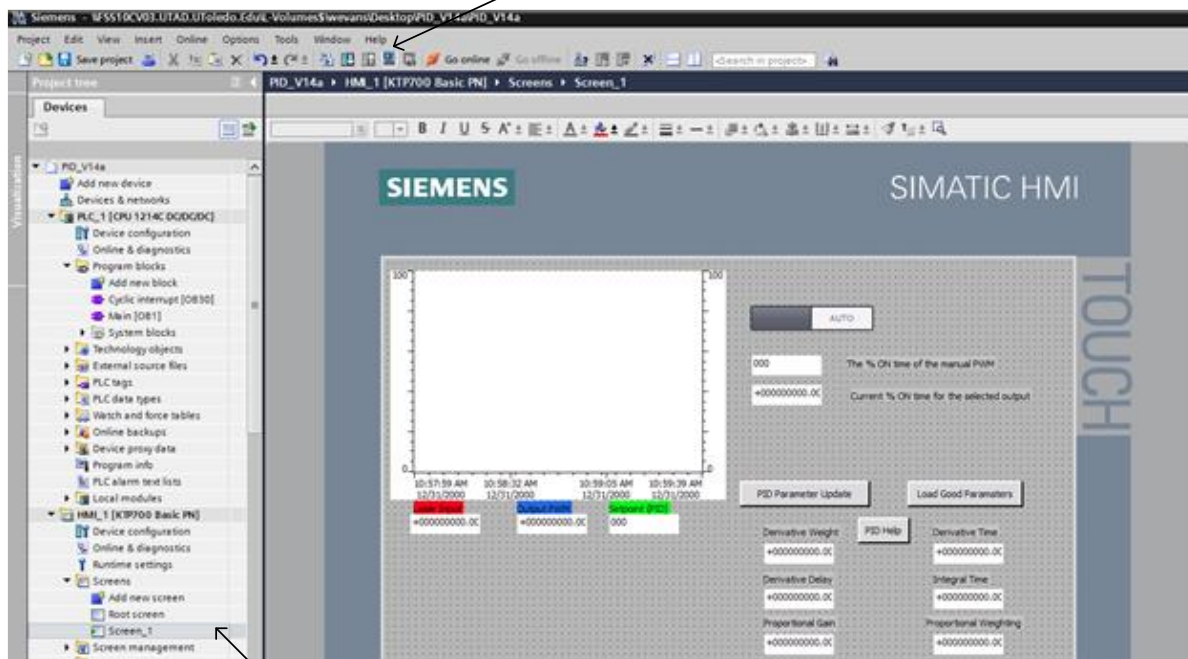
We need to find the variables that need to be written to the historical data. Click on Laser Input and get the tag Laser_Percent. This tag is one to be written to the historical data logger.
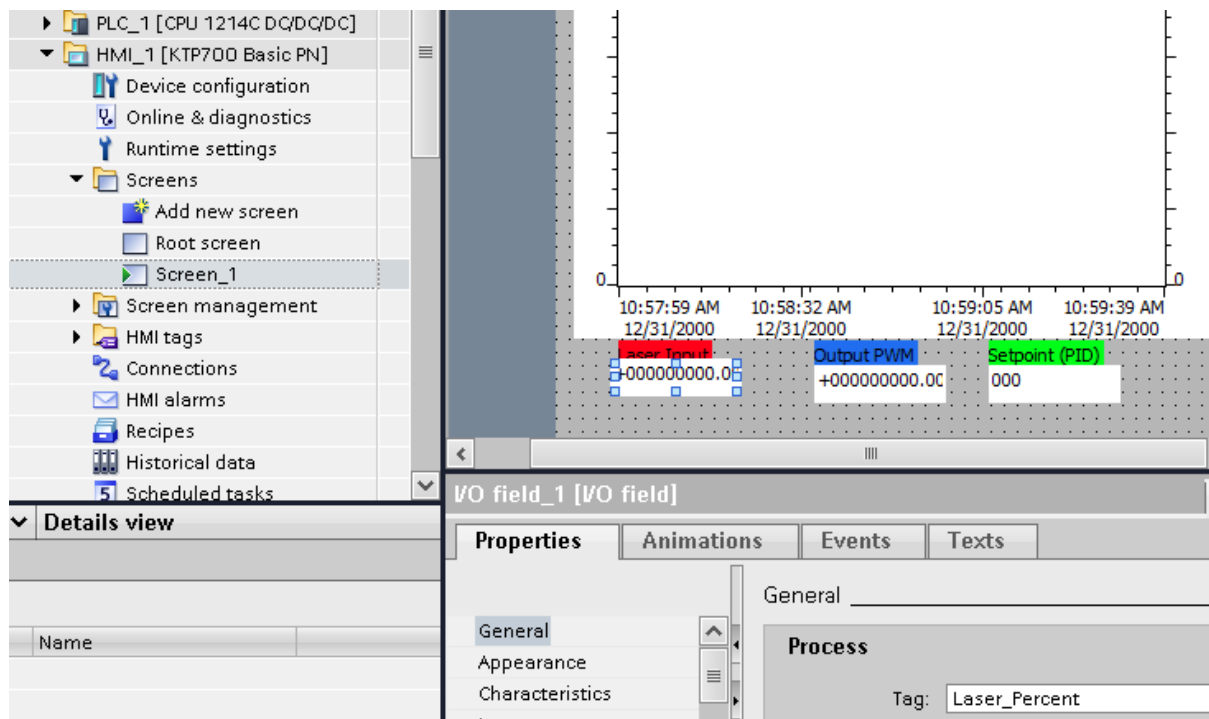
Next, go to the Historical data tag under HMI tags. Set the variables from the HMI screen above that are to be saved. Fill in the appropriate fields and start the historical logger. This circular file will contain the data from the analog data saved. Then run the program and run the Historical data logger. It is a circular file and will wrap around after the table fills up.