



## Silver Series OIT Sample Projects

### Using the EZ-ZONE<sup>®</sup> RM 800x480 Ethernet Sample Project

These instructions assume you have obtained an archive (\*.zip) containing a sample project and opened these instructions from within that archive.

#### Installing the Sample Project

To install and inspect the sample project:

1. Copy the project file (the file with the ".mtp" extension) and the compiled project (the file with the \*.xob extension) from the archive to the project directory in EasyBuilder5000. Typically "C:\Watlow\EZ5000\project".
2. If desired, copy the sample application description (the file with the .pdf extension) to your desktop or other suitable location.
3. Launch EasyBuilder5000 (Start/All Programs/Watlow/EZware-5000/EasyBuilder5000)
4. From the File menu choose Open.
5. Locate and select the sample project and click Open.

Now you can work with the sample project just as you would any other project. You can examine it to see how things were done, compile and download it to an OIT.

#### Hardware Configuration

This sample project is designed to communicate as indicated below. In order to use this with your hardware, you may need to adjust the settings in your controller or in the System Parameters Settings dialog in EasyBuilder5000.

<b>Controller</b>	EZ-ZONE RM		
<b>Model</b>	RMC1_____ or equivalent RMA - 3 - _____ or equivalent		
<b>Address</b>	1 (RMC) 192.168.240.2 (RMA Port 2)		
<b>Protocol</b>	Modbus TCP	<b>Word Order</b>	Low High
		<b>Gateway Offset 1</b>	0

<b>OIT</b>	TS00-0070-0000
<b>Interface</b>	Ethernet

## Things to Observe

Note the following features in the sample project.

### Common Window (Window 4):

- The mouse cursor (pointer) is disabled with a Set Bit object that sets the System Tag LB-9018 to On when the window opens.
- The navigation buttons (Function Keys) that appear on every full screen window are actually only on the common window.
- The “PW Setup” Function Key is set to be invisible on its Security Tab except to users that have access to Class C objects.

### Initial Window (Window 10):

- Some of the objects such as the Meter Display for the Heat Power and the Option List for the Control Mode take up more space in the screen designer than they occupy when the window is running on the OIT. This can limit where they are placed, i.e. they cannot be placed so close to the bottom of the screen that the invisible parts would extend off the screen. However, the space can be used for other objects. Note how the top of the Bar Graph is in the space the Option List covers when it is open.

### Recipes (Window 11):

- This is a pop-up window that appears when the *Select Recipe* Function Key on the Recipe Manager window is pressed.
- The five ASCII Display objects list the names users have assigned to recipes. These names are stored in memory locations RW-10, RW-30, RW-50... to ...RW-490, but only the names of five recipes are shown at a time. Which recipe names are shown depends on the setting of address index 1.
- The up and down arrow keys are Set Word objects that increment and decrement the system word LW-9201, address index 1, from 0 to 400 in steps of 100. This is used to scroll through the list of recipe names five recipes at a time. When address index 1 is set to 0 the five ASCII Display objects show the recipe names stored at RW-10, RW-30, RW-50, RW-70 and RW-90. When the user presses the down arrow which increments address index 1 by 100 the list shows the recipe names stored at RW-110, RW-130, RW-150, RW-170 and RW-190. Pressing the down arrow again shows the next five recipe names in the list.
- The up and down arrow keys are prevented by limits on their General tabs from setting the value of address index 1 less than 0 or greater than 400.
- The up and down arrow keys also set a bit (LB-550). When that bit is set, the PLC Control object runs a macro called *Move Slider* that sets LW-1001. This word determines the position of the Animation Object that appears to be a scroll bar and indicates where you are in the list of recipe names. See the PLC Control object by opening it from the *Object* menu and the Macros from the *Tools* menu.
- Each of the five *Edit* buttons (Set Bit objects) runs a corresponding macro that sets the address 0 index such that the corresponding recipe is selected.

### Trend (Window 12):

- The vertical scale is not part of the Trend Display object. Instead it is a Scale Object and six Text Objects that have been placed next to the trend.
- On the Trend Object's General tab you see that the trend displays data from the *Data Sampling Object index* "0.Loop 1". You can see the Data Sampling Object by opening it from the *Objects* menu.
- If you inspect the Data Sampling object, you will see that it is reading four local words starting as LW-200, an address internal to the OIT. This is because a trend and a data log can include data from only one Data Sampling object and a Data Sampling object can contain only contiguous data. In order to graph set point and process variable on the same graph, *Data Transfer (Time-Based)* objects are used to copy the data from the controller to contiguous memory locations in the OIT. Four words in memory are used because the set point and process variable are floating point numbers and therefore require 32-bits of memory each which requires two of the OIT's local words each to store. This method can just as easily be used to create a log or graph of data from multiple controllers.
- On the Trend Object's *Channel* tab, the scale for each of the two channel's (0 and 1) is set to 0 to 250. It is also possible to have a user set scale by selecting the Dynamic limits option.

### Recipe Manager (Window 13):

- The recipe consists of the five controller parameters plus the user-set recipe name.
- The recipes are stored in *recipe word* (RW) memory which is saved to non-volatile (flash) memory periodically by the OIT (every 5 minutes). So it is important to wait for this to happen after editing a recipe and before turning off power to the OIT if you want the recipes to be stored in the recipe memory.
- The controller's actual settings are displayed for reference only. It is not necessary to have them on the screen in order for the recipe functions to work.
- The controller settings are copied from the controller to recipe memory by the five *Data Transfer (Trigger-Based)* objects that also serve as labels next to the Numeric Display objects that show the actual controller values. The trigger for the data transfers is a bit in the OIT (LB-500) that is momentarily set when the user clicks the *Save as Selected Recipe* button, a Set Bit object.
- Similarly there are five Data Transfer (Trigger-Based) objects that copy values from the selected recipe to the controller when the *Load Recipe into Controller* button momentarily sets their trigger (LB-501).
- This example allows for 25 recipes stored in RW memory in the OIT. Recipe 1 is stored in addresses 0 to 19, recipe 2 in addresses 20 to 39 and so on.
- Which recipe memory locations are saved to or loaded is determined by the setting of *address index 0* which is stored in system tag LW-9200. Note that the five Data Transfer (Trigger-Based) objects and the five Numeric Display objects associated with the *Selected Recipe Settings* each has its Index Register option set and Index 0 selected on the General tab. That means the value stored in the address index is added to the address entered for the object. So the recipe's set point will be stored

at address RW-0 when Index 0 is set to zero, but it will be stored at address RW-20 when the value in Index 0 is 20.

- The *Previous Recipe* and *Next Recipe* buttons are Set Word objects that increment the recipe memory index (Index 0) up and down by 20 so that values can be stored in any of the recipes. These buttons are limited to setting the index to multiples of 20 from 0 to 480 which means there are 25 sets of address available for recipes and therefore 25 recipes.
- The *Select Recipe* Function Key pops-up the Recipes window which can also be used to select a recipe.

#### **Test Passwords (Window 14):**

- The three Numeric Input objects provide examples of different levels of protection as indicated by the text next to each. See the settings on the Security tab for each to see how the objects are configured to behave.
- From the Edit menu select System Parameters and look at the Security tab to see how the three users were enabled and assigned default passwords and permission to access the object classes (A to F).
- The user passwords are stored in system tags that *can be* displayed (as they are here to minimize lost password frustration in a demo) but typically *should not be*.

#### **Log In (Window 15):**

- This window pops-up when you push the *Log In* button.
- A *Multi-State Switch* is used to set the User Number System tag (LW-9219).
- A Numeric Input object with the mask option set on its Numeric Format tab is used to allow users to input the password corresponding to the selected user into the Password System tag (LW-9220).
- The Bit Lamp object indicates a failed attempt to log in if the user enters the wrong password for the selected user. The failed attempt is flagged by the Password Error System Tag (LB-9060).

#### **Change Passwords (Window 16):**

- This window pops-up when you push the *Change Passwords* button. Note this button is on the Common window, but it is only visible when User 3 is logged in.
- The Numeric Input objects are used to allow an authorized user to set any user's password but not to view them. The user passwords are stored in System Tags (LW-9500, LW-9502, and LW-9504 for users 1, 2 and 3).
- The Save Changes button is a Set Bit object that momentarily sets the Update Password System Tag (LB-9061) which saves any edits made to the passwords in the Numeric Inputs.