TECHNICAL NOTE

Maple Model(s)	**Title**	**TN5103**
Graphic HMI	Recipe Macros	

P/N: 0907-5103
Rev. 00 Date: 05/06/2015

Summary

This technical note covers concepts used when working with the recipe database features in EZwarePlus. For more information about recipes and handling the recipe database files, see chapter 15 in the EZwarePlus programming manual.

Solution

NOTE: This section uses the HMI demo applications that are installed in the EZwarePlus Projects folder. (HMI5xxxxdemo.emtp)

In the demo projects, the Recipe View object is used to display and select recipe records for one recipe in the recipe database.

In the demo applications, ASCII and numeric objects on the Edit Recipe popup are assigned to the corresponding recipe elements. The value of the recipe element for the recipe record selected by the recipe view object will be displayed in these objects and can be altered using the ASCII and numeric input objects.

Once a new value is entered in one of the numeric objects, the value is in a holding register and can be stored to the recipe database by writing a 2 to the recipe command register. In the sample project, this is done with a Set Word object stacked on top of the Save Changes button on the Edit Recipe popup window.

If the Load Recipe button is pressed, the data in the holding registers is transferred to the “PLC registers”. In the demo application, the PLC registers are simulated with Local Words starting at LW1720.

Search Recipes

The ASCII object under the Search Recipes header is configured to accept input so a recipe name can be entered. In this example, the recipe name is entered in an HMI tag called RCP_SearchString (RW1700). The recipe search macro is started with a PLC Control Object (Objects > PLC Control Object) triggered by RCP_StartSearch (LB1705).

Once a name is entered in RCP_SearchString and bit RCP_StartSearch is true, the macro compares the name entered in RCP_SearchString against the Recipe “Flavor” element. The recipe element to be compared can be altered in line 20 of the macro.

If the recipe name is found, the recipe selection value is changed to the corresponding recipe record and is highlighted on the Recipe View object. The recipe search macro will be disabled.

To build this function into an application, copy the following macro code and paste it in a macro development area (Tools > Macros > New) with no alterations then compile. There should be zero errors. If any errors are presented, reattempt this process.

Create a PLC control object that will execute the macro when a bit is set. Create an ASCII object that will accept input to enter the recipe record name to search and a toggle switch object to trigger the PLC Control Object that runs the search macro.

Some names in the recipe search macro will need to be altered depending on the names given to the recipe and recipe elements. Use the demo application as an example.

RCP_FindRecipe

```
//RCP_FindRecipe - Searches through the recipe database to find the string entered in the search box

//and, if found, select the desired recipe

macro_command main()

  //declarations

  short i = 0

  short count      // number of recipes

  char RName[20]   // Buffer for recipe name

  char FName[20]   // Buffer for search string

  bool result      // 1 if found 0 otherwise

  bool OFF = 0


  //Get number of recipes and search string

  GetData(count, "Local HMI", RECIPE, "IceCream.Count")

  GetData(FName[0], "Local HMI", RW, 1700, 10)
  //Loop through the recipies and compare strings
  while i < count


    RecipeGetData(RName[0], "IceCream.Flavor", i)              //Read in the ith recipe name
    result = StringCompareNoCase(FName[0], RName[0])           //compare
    if result == true then //found a match
      SetData(i, "Local HMI", RECIPE, "IceCream.Selection")   //set selection
      i = count                                    // exit the loop
    end if
    i = i + 1                                      //move to next recipe
    DELAY(10)
  wend


  SetData(OFF, "Local HMI", "RCP_StartSearch", 1)              //reset search flag


end macro_command
```

Update Recipe Names in the Option List

The Edit Recipe, Delete and Add New buttons trigger the notification bit (LB1700) when pressed. The bit can be configured to be set on before or after writing the initial value dependent on the function to be accomplished.

When LB1700 is triggered, a PLC Control Object executes the RCP_GetRecipeNames macro (code is below) to retrieve the names from the recipe records and places those names in a series of registers in local memory starting at RW1750.

The Option List is configured to get the item data from an item address. The item address is configured to get the recipe names from RW1750 with a length of 10 words per element just as the name element of the recipe is configured.

To build this function into an application, copy the following macro code and paste it in a macro development area (Tools > Macros > New) with no alterations then compile. There should be zero errors. If any errors are presented, reattempt this process.

Create a PLC control object that will execute the macro when a bit is set. Create edit, delete buttons and enable the notification bit. See the demo application for examples on the settings of these bits.

Create an Option List object. Set the Source of Item Data to Item Address. Configure the Item Address location to RW1750 (or the register chosen to place the recipe names).

RCP_GetRecipeNames

```
//RCP_GetRecipeNames - Updates the Option List on the recipe screen with the names of the

//recipes. This macro should be called whenever the Recipe database

//modified. e.g. when a recipe is added, deleted or the name is modified.

//The recipe names are saved in LW memory at the address specified by addr.

macro_command main()

  //declarations

  short i

  short addr = 1750  // starting address of buffer to store recipe names

  short count        // number of recipes

  char data[20]      // buffer to store a recipe name

  bool OFF = 0
  bool ON = 1


  // read in recipe count

  GetData(count, "Local HMI", RECIPE, "IceCream.Count")


  // Loop through recipe data base and copy each name

  // to LW memory

  for i = 0 to count-1


    RecipeGetData(data[0], "IceCream.Flavor", i)

    SetData(data[0], "Local HMI", RW, addr, 20)


    addr = addr + 10 //move to buffer for next recipe

  next i


  //update Object List parameters

  SetData(count, "Local HMI", LW, 1702, 1)

  SetData(ON, "Local HMI", LW, 1701, 1)               // set Object List's update flag

  SetData(OFF, "Local HMI", "RCP_UpdateRecipeList", 1)     // reset Macro execution flag


end macro_command
```