

Machine recipe

Many machines have a big set of parameters, depending on the kind of job that you want to do. Try to imagine cooking something: you will need some specific ingredients and the instructions to do a determined recipe.

In the automation world it is the same: a system like a milling machine may have a huge set of parameters such as speed, acceleration, jerk, temperature limit and quantity of cooling material used. Every time you want your machine to do a job on a specific material (such as wood, stone, silicon, plastic....), you would like to load a file with related set of parameter, like reading a recipe.

For this reason, Robox provides a specific OB, called **rc_recipe**, useful to manage data with functions like save, load or delete. This exercise is meant to learn how to use it and make a good HMI project to control it.

1. RDE project with rc_recipe

Open RDE and make a project using a rc_recipe OB. It is not necessary to have any axes, just a task to write necessary code to handle this OB. Read provided manual to learn more. Your project must make the initialization following these steps, where “recipe” is an instance of rc_recipe OB:

```
; data dimension  
recipe.init(3000, 0x00000001)  
; Add the recipe title NVSR(1), used for title  
recipe.append(recipe.RC_RECIPE_R_NV, recipe.RC_RECIPE_T_STR, 1, 1)  
; Add (500 nvrr parameter)  
recipe.append(recipe.RC_RECIPE_R_NV, recipe.RC_RECIPE_T_REAL, 1, 500)
```

In addition to this, your recipe OB must set a title, a prefix to save files in /recipe/ folder of your flash and each saved file must be called “prog_XXX”, where “XXX” is a progressive number. Set the total recipes number to 100.

In your task you must write a function to handle the recipe, able to read some commands such as file_get_title, file_load, file_save and file_delete. You can use a R(n) register and give to each bit a meaning; you will write these bits by pressing buttons on your HMI.

2. Robox Display Tool (RDT)

Open RDT and make a project to handle your recipe: a page must be a main recipe handler, as shown in **picture 1**, below.



Picture 1 – Example of main recipe page.

Put some buttons to select, load, save and delete your recipes. To make a nice HMI, remember to use icons and not text. This is always an important principle to develop a modern and friendly HMI.

Remember to ask confirmation to all commands before executing them, to avoid button pressed by mistake. In the lower side of this slide, an output string is requested, to print what is happening, such as “load done”, “file correctly saved” or to print any error. It is also requested to show current recipe number and title.

Another page will show the recipe list, with progressive numbers and titles. This can be as shown in **picture 2**. Remember that the user has up to 100 recipes available, so use some buttons to allow user to increment or decrement the recipe number. For example, this page shows recipes from 1 to 20, if I press the right arrow, I’ll see 21 to 40, then 41 to 60 and so on. It is suggested to handle this in the task too, so in the HMI just give a command to some register bits.



Picture 2 – recipe list.

Note: the HMI must never slow down too much the system, so try always to avoid any alarm such as “task reduced frequency” or “task rule excessive length”. Read titles and data only if requested, otherwise give a “return” command to your functions.

3. Debug

Compile and load your RDE project in your controller, then reboot and connect it to your HMI. Compile and load your RDT project in your HMI. If you do not have an HMI device, you can use RTM directly on your PC. Try to use your HMI to move among the various slides and use all functions: load, save and delete.