

# Custom Menu

Version 1.1

Written by:  
Curt Binder  
[curt.binder@gmail.com](mailto:curt.binder@gmail.com)  
<http://curtbinder.info/>



## Table of Contents

Conventions.....	3
Requirements.....	3
Introduction.....	4
Enable Custom Menu.....	4
Defining the Menu.....	4
Creating Custom Functions.....	5
Initializing Custom Menu.....	6
Complete PDE file.....	6
Complete List of Custom Menu Options.....	8
Available Functions.....	9
Examples.....	10
Turn on Specific Ports.....	10
Turn On & Off Port.....	11
Immediate Display of Menu.....	11
Wait for Joystick.....	11
Final Note.....	12

# Conventions used in this guide

## Section Headers – this is what a section header looks like.

This indicates a new section. This is for quick reference to the sections.

*Notes – special notes appear like this.*

Any special notes about the code will appear like this.

```
Code Segments - all code segments look like this.
```

Code segments that are capable of being used are in a monospace type of font for easy reading.

## Requirements

*Requirements for this guide*

- *Dev libraries 0.8.5.17 or later*
- *Some programming knowledge*
- *Patience*

# Introduction

This guide will serve as a basis for creating a custom menu for your Reef Angel controller. The list of requirements are at the start of this guide.

The first thing you will need to do is enable custom menu support in the features file. As of the latest release of RAGen (1.0.4), this feature is not included so it will have to be added manually.

First we need to locate the libraries folder. The libraries folder is usually located in the following folder:

Operating System	Libraries Folder Location
Windows XP	My Documents\Arduino\libraries
Windows Vista or 7	Documents\Arduino\libraries
Linux	~/arduino/libraries
OS X	~/Documents/Arduino/libraries

Once we have that folder located, you need to open it up inside your File Explorer. Inside that folder, you will see lots of folders that begin with the ReefAngel prefix. You need to find (and open) the folder called **ReefAngel\_Features**. Inside that folder, there is another file called **ReefAngel\_Features.h**. You need to open up that file inside your favorite text editor. You will need to add the following lines to this file:

```
#define CUSTOM_MENU
#define CUSTOM_MENU_ENTRIES 7
```

Once you add these lines to the file, you can save and close it. The second line tells how many menu entries will be in your menu. For copying the Simple Menu, you will need 7 entries. You can change this to match your menu.

For the example we are going to create, you must have the Version Menu and Date / Time setup enabled in the features file as well. Make sure your features file also contains these lines:

```
#define DateTimeSetup
#define VersionMenu
```

Now that we have the Custom Menu enabled with the libraries, we need to add the extra functionality to the PDE file. We will start by adding in the functions to display the menu and mimic the Simple Menu.

## Defining the Menu

The first thing we did was define the menu itself. This is handled in the following block of code:

```
#include <avr/pgmspace.h>
// Create the menu entries
prog_char menu1_label[] PROGMEM = "Feeding";
prog_char menu2_label[] PROGMEM = "Water Change";
prog_char menu3_label[] PROGMEM = "ATO Clear";
prog_char menu4_label[] PROGMEM = "Overheat Clear";
```

```

prog_char menu5_label[] PROGMEM = "PH Calibration";
prog_char menu6_label[] PROGMEM = "Date / Time";
prog_char menu7_label[] PROGMEM = "Version";
// Group the menu entries together
PROGMEM const char *menu_items[] = {
menu1_label, menu2_label, menu3_label,
menu4_label, menu5_label, menu6_label,
menu7_label
};

```

The first grouping is where we create the menu entries. You can change the text inside the quotation marks to be whatever you want it to be. The menu labels are in order from top down.

*Note: The text cannot exceed 20 characters.*

After the menu entries are created, we have to group them together to make it easy to operate. That is handled after the “Group the menu entries together” line. We put the labels in order separated by a comma.

Now we have our list of menu entries stored in the variable called “menu\_items”. This will be used later.

## Creating Custom Functions

Once the menu entries are created, we now need to tell the controller the functions/code that is to be run when we click / select a menu entry. This code is handled in specific functions. So we create the functions that the controller is looking for. Those functions are here:

```

void MenuEntry1()
{
    ReefAngel.FeedingModeStart();
}
void MenuEntry2()
{
    ReefAngel.WaterChangeModeStart();
}
void MenuEntry3()
{
    ReefAngel.ATOClear();
    ReefAngel.DisplayMenuEntry("Clear ATO Timeout");
}
void MenuEntry4()
{
    ReefAngel.OverheatClear();
    ReefAngel.DisplayMenuEntry("Clear Overheat");
}
void MenuEntry5()
{
    ReefAngel.SetupCalibratePH();
    ReefAngel.DisplayedMenu = ALT_SCREEN_MODE;
}
void MenuEntry6()
{

```

```

    ReefAngel.SetupDateTime();
    ReefAngel.DisplayedMenu = ALT_SCREEN_MODE;
}
void MenuEntry7()
{
    ReefAngel.DisplayVersion();
}

```

For simplicity, the functions are named based on what menu entry they are associated with. So menu entry 1 (menu1\_label), calls MenuEntry1 function. The code you want to be run when you select the first menu entry is contained inside this function. Menu entry 6 (menu6\_label) calls MenuEntry6 function and so forth.

## Initializing Custom Menu

The last thing we have to do is to initialize the menu for use on the controller. This is handled in the setup function right after we initialize the controller.

```

void setup()
{
    ReefAngel.Init(); //Initialize controller
    ReefAngel.InitMenu(pgm_read_word(&(menu_items[0])), SIZE(menu_items));

    // .. other code here
}

```

The InitMenu line is what initializes your menu. If you just modify the existing code (recommended), you will not have to change anything with this line. You just have to make sure you add it in.

## Complete PDE file

Continuing on with this example, here is what the default PDE file (RACustomMenu.pde) will look like with this code added.

```

// RACustomMenu.pde
//
// This version designed for v0.8.5 Beta 17 and later

#include <ReefAngel_Features.h>
#include <ReefAngel_Globals.h>
#include <ReefAngel_Wifi.h>
#include <Wire.h>
#include <OneWire.h>
#include <Time.h>
#include <DS1307RTC.h>
#include <ReefAngel_EEPROM.h>
#include <ReefAngel_NokiaLCD.h>
#include <ReefAngel_ATO.h>
#include <ReefAngel_Joystick.h>
#include <ReefAngel_LED.h>
#include <ReefAngel_TempSensor.h>

```

```

#include <ReefAngel_Relay.h>
#include <ReefAngel_PWM.h>
#include <ReefAngel_Timer.h>
#include <ReefAngel_Memory.h>
#include <ReefAngel.h>

// Custom Menu Code
#include <avr/pgmspace.h>
prog_char menu0_label[] PROGMEM = "Feeding";
prog_char menu1_label[] PROGMEM = "Water Change";
prog_char menu2_label[] PROGMEM = "ATO Clear";
prog_char menu3_label[] PROGMEM = "Overheat Clear";
prog_char menu4_label[] PROGMEM = "PH Calibration";
prog_char menu5_label[] PROGMEM = "Date / Time";
prog_char menu6_label[] PROGMEM = "Version";
PROGMEM const char *menu_items[] = {
menu0_label, menu1_label, menu2_label,
menu3_label, menu4_label, menu5_label,
menu6_label//, menu7_label, menu8_label
};

void MenuEntry1()
{
    ReefAngel.FeedingModeStart();
}
void MenuEntry2()
{
    ReefAngel.WaterChangeModeStart();
}
void MenuEntry3()
{
    ReefAngel.ATOClear();
    ReefAngel.DisplayMenuEntry("Clear ATO Timeout");
}
void MenuEntry4()
{
    ReefAngel.OverheatClear();
    ReefAngel.DisplayMenuEntry("Clear Overheat");
}
void MenuEntry5()
{
    ReefAngel.SetupCalibratePH();
    ReefAngel.DisplayedMenu = ALT_SCREEN_MODE;
}
void MenuEntry6()
{
    ReefAngel.SetupDateTime();
}

```

```

    ReefAngel.DisplayedMenu = ALT_SCREEN_MODE;
}
void MenuEntry7()
{
    ReefAngel.DisplayVersion();
}

void setup()
{
    ReefAngel.Init(); //Initialize controller
    ReefAngel.InitMenu(pgm_read_word(&(menu_items[0])), SIZE(menu_items));

    // Ports that are always on
    ReefAngel.Relay.On(Port8);
}

void loop()
{
    ReefAngel.ShowInterface();

    // Specific functions
    ReefAngel.SingleATOLow(Port6);
    ReefAngel.Relay.DelayedOn(Port1, 1);
    ReefAngel.StandardLights(Port2);
    ReefAngel.MHLights(Port3);
    ReefAngel.Wavemaker1(Port4);
    ReefAngel.Wavemaker2(Port5);
    ReefAngel.StandardHeater(Port7);
}

```

The code added is the same code that displays the Simple Menu inside the libraries. The only difference is that now, we can change and update it without having to modify the libraries. This will allow you to upgrade the libraries without losing your changes.

## Complete List of Custom Menu Options

There can be a maximum of 9 menu entries. You must specify the number of menu entries inside the Features file (see above).

Here is a list of the Menu Entry options:

Menu Entry	Label	Function
Item 1	prog_char menu1_label[] PROGMEM = "Item 1";	void MenuEntry1() { }
Item 2	prog_char menu2_label[] PROGMEM = "Item 2";	void MenuEntry2() { }
Item 3	prog_char menu3_label[] PROGMEM = "Item 3";	void MenuEntry3() { }



Item 4	prog_char menu4_label[] PROGMEM = "Item 4";	void MenuEntry4() { }
Item 5	prog_char menu5_label[] PROGMEM = "Item 5";	void MenuEntry5() { }
Item 6	prog_char menu6_label[] PROGMEM = "Item 6";	void MenuEntry6() { }
Item 7	prog_char menu7_label[] PROGMEM = "Item 7";	void MenuEntry7() { }
Item 8	prog_char menu8_label[] PROGMEM = "Item 8";	void MenuEntry8() { }
Item 9	prog_char menu9_label[] PROGMEM = "Item 9";	void MenuEntry9() { }

Then here's the complete line to group all the menu entries together:

```
PROGMEM const char *menu_items[] = {
menu1_label, menu2_label, menu3_label,
menu4_label, menu5_label, menu6_label,
menu7_label, menu8_label, menu9_label
};
```

## Available Functions

Here is a list of the functions inside libraries that you can use.

```
ReefAngel.DisplayVersion();
ReefAngel.ClearScreen(byte Color);
ReefAngel.PWMExpansion(byte cmd, byte data);
ReefAngel.PWMSetPercent(byte p);
ReefAngel.FeedingModeStart();
ReefAngel.WaterChangeModeStart();
ReefAngel.ATOClear();
ReefAngel.OverheatClear();
ReefAngel.SetupCalibratePH();
ReefAngel.SetupDateTime();
```

If you are clearing the screen to display your own screen/text, then you will want to set the DisplayedMenu to be in the Alternate Screen mode (ALT\_SCREEN\_MODE), so when you exit the function, the menu will display as desired and operate as expected. You MUST do this for the SetupCalibratePH() and SetupDateTime() functions to work properly. If you do not, you will have to press the joystick button 2 times to return to the menu and the display to work as desired.

If you want your custom function to return to the main screen when it exits, then you must set the DisplayedMenu to be RETURN\_MAIN\_MODE. This will handle the necessary cleanup for returning to the main screen.

Even though these are the only "pre-made" functions, that does not mean you can't create your own functionality inside the Menu Entries to get the job done. You do not have to use the existing functions if you

do not want to. You can look in the libraries and use the code that is there. One common use for that would be to add in a setup screen. The setup menus are not available / included for use with the Custom Menu. If you want to use them, you will have to mimic them inside your own functions.

You can also use all the functions that are available in the Custom Main Screen guide.

If you want to control any specific relays, you can use these functions.

```
ReefAngel.Relay.On(byte ID);
ReefAngel.Relay.Off(byte ID);
ReefAngel.Relay.AllOn();
ReefAngel.Relay.AllOff();
ReefAngel.Relay.Toggle(byte ID);
ReefAngel.Relay.Write();
```

*Note: If you call any of the functions, you MUST call the Write() function to make the relays actually change their status (on/off).*

## Examples

The next section will contain various examples that can be used on your controller or as a reference for creating your own custom menu.

### Turn on Specific Ports

This example will use a custom entry to turn on ports 2, 3, & 4 when we select the entry. It will then wait until you press the joystick button and then turn off those ports. Then it will return to either the main screen or the menu.

It will NOT monitor the rest of your tank while the ports are on. Your existing relays will stay in their current state. So if Port 1 was already on, it will stay on until you exit your function and the main controller software has a chance to alter the port status as desired.

This is set to work when the 3<sup>rd</sup> menu item is selected.

```
void MenuEntry3()
{
  // Clear the screen
  ReefAngel.ClearScreen(DefaultBGColor);
  // Display some text on the screen
  ReefAngel.LCD.DrawText(DefaultFGColor, DefaultBGColor, 5, 10, "My Lights
On");
  // Turn on the ports
  ReefAngel.Relay.On(Port2);
  ReefAngel.Relay.On(Port3);
  ReefAngel.Relay.On(Port4);
  ReefAngel.Relay.Write();
  bool bDone = false;
  do
  {
    // wait indefinitely for a button press
    delay(200);
```

```

    if ( ReefAngel.Joystick.IsButtonPressed() )
    {
        bDone = true;
    }
} while ( ! bDone );
// Turn off the ports we turned on
ReefAngel.Relay.Off(Port4);
ReefAngel.Relay.Off(Port3);
ReefAngel.Relay.Off(Port2);
ReefAngel.Relay.Write();
// Tell the controller to cleanup and return to the main screen
ReefAngel.DisplayedMenu = RETURN_MAIN_MODE;
// To return to the menu instead, comment out the above line and use
// this line instead
//ReefAngel.DisplayedMenu = ALT_SCREEN_MODE;
}

```

## Turn On & Off Port

This example will add an entry to turn on and turn off the return pump that gets turned on at controller startup. There are 2 ways you can handle this:

1. Turn the port on/off and immediately display the menu again.
2. Turn the port on/off and wait for the joystick to be pressed before displaying the menu again.

We will be adding entries to items 6 & 7 on the menu for this example. Here are the menu entry labels:

```

prog_char menu6_label[] PROGMEM = "Turn ON Return";
prog_char menu7_label[] PROGMEM = "Turn Off Return";

```

## Immediate Display of Menu

This will display the menu immediately after selecting the option.

```

void MenuEntry6()
{
    ReefAngel.Relay.On(Port8);
    ReefAngel.Relay.Write();
    ReefAngel.DisplayedMenu = ALT_SCREEN_MODE;
}
void MenuEntry7()
{
    ReefAngel.Relay.Off(Port8);
    ReefAngel.Relay.Write();
    ReefAngel.DisplayedMenu = ALT_SCREEN_MODE;
}

```

## Wait for Joystick

This will wait for the joystick to be pressed before displaying the menu.

```

void MenuEntry6()
{
    ReefAngel.Relay.On(Port8);
    ReefAngel.Relay.Write();

```

```
    ReefAngel.DisplayMenuEntry("Turn ON Return");  
}  
void MenuEntry7()  
{  
    ReefAngel.Relay.Off(Port8);  
    ReefAngel.Relay.Write();  
    ReefAngel.DisplayMenuEntry("Turn Off Return");  
}
```

Either way will work as desired, it just depends on how you want the controller to display.

## Final Note

There are lots of ways the custom menu can be used to perform tasks on your tank. You will just need to experiment to figure out what needs to be done. There is room for improvement with the way the controller responds during your custom functions. There may be changes made in future releases as improvements are made.

Also remember, if you are inside your own function, the controller will not be processing any of the functions inside loop(). So the heaters/fans will not turn on/off with changes in temperatures. The lights will not turn on/off on their schedule. The ATO will not turn on/off. The controller will not respond to WIFI requests. (To respond to the WIFI requests, you will just need to call pingSerial() inside your code).

If you get stuck, ask questions in the forums (<http://forum.reefangel.com/>).

Happy Coding!

curt