Custom Main Screen

Version 1.3

Written by:
Curt Binder
curt.binder@gmail.com
http://curtbinder.info/

Table of Contents

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 0.9.0 or later*
- *Some programming knowledge*
- *Patience*

Introduction

This guide will serve as a basis for creating a custom main screen 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 main screen support in the features file. RAGen is no longer used. The Wizard is recommended to create INO files.

Open up your PDE/INO file (inside Arduino) and add in the following code above the setup() function:

```
////// Place global variable code below here
void DrawCustomMain()
{
     // the graph is drawn/updated when we exit the main menu &
     // when the parameters are saved
     ReefAngel.LCD.DrawDate(6, 112);
     pingSerial();
#if defined DisplayLEDPWM && ! defined RemoveAllLights
     ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params,
ReefAngel.PWM.GetDaylightValue(), ReefAngel.PWM.GetActinicValue());
#else // defined DisplayLEDPWM && ! defined RemoveAllLights
     ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params);
#endif // defined DisplayLEDPWM && ! defined RemoveAllLights
     pingSerial();
     byte TempRelay = ReefAngel.Relay.RelayData;
     TempRelay &= ReefAngel.Relay.RelayMaskOff;
     TempRelay |= ReefAngel.Relay.RelayMaskOn;
     ReefAngel.LCD.DrawOutletBox(12, 93, TempRelay);
}

void DrawCustomGraph()
{
     ReefAngel.LCD.DrawGraph(5,5);
}
////// Place global variable code above here
```

Continuing on with this example, here is what the default INO file (RAMenus.ino) will look like with this code added.

```
// RAMenus.ino
//
// This version designed for v0.9.9 and later

#include <ReefAngel_Features.h>
#include <Globals.h>
#include <RA_Wifi.h>
#include <Wire.h>
#include <OneWire.h>
#include <Time.h>
#include <DS1307RTC.h>
```

```
#include <InternalEEPROM.h>
#include <RA_NokiaLCD.h>
#include <RA_ATO.h>
#include <RA_Joystick.h>
#include <LED.h>
#include <RA_TempSensor.h>
#include <Relay.h>
#include <RA_PWM.h>
#include <Timer.h>
#include <Memory.h>
#include <ReefAngel.h>

////// Place global variable code below here
void DrawCustomMain()
{
     // the graph is drawn/updated when we exit the main menu &
     // when the parameters are saved
     ReefAngel.LCD.DrawDate(6, 112);
     pingSerial();
#if defined DisplayLEDPWM && ! defined RemoveAllLights
     ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params,
ReefAngel.PWM.GetDaylightValue(), ReefAngel.PWM.GetActinicValue());
#else // defined DisplayLEDPWM && ! defined RemoveAllLights
     ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params);
#endif // defined DisplayLEDPWM && ! defined RemoveAllLights
     pingSerial();
     byte TempRelay = ReefAngel.Relay.RelayData;
     TempRelay &= ReefAngel.Relay.RelayMaskOff;
     TempRelay |= ReefAngel.Relay.RelayMaskOn;
     ReefAngel.LCD.DrawOutletBox(12, 93, TempRelay);
}

void DrawCustomGraph()
{
     ReefAngel.LCD.DrawGraph(5, 5);
}
////// Place global variable code above here

void setup()
{
     ReefAngel.Init(); //Initialize controller

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

void loop()
{
     // Specific functions
     ReefAngel.StandardATO(Port1);
     ReefAngel.StandardLights(Port2);
     ReefAngel.MHLights(Port3);
     ReefAngel.StandardFan(Port4);
     ReefAngel.Wavemaker1(Port5);
     ReefAngel.Wavemaker2(Port6);
     ReefAngel.StandardHeater(Port7);

     ReefAngel.ShowInterface();
}
```

The code added is the same code that displays the main screen 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.

Now that we have the finished version, let's examine what each section and item does. First off, let's discuss the functions in general DrawCustomMain() and DrawCustomGraph()

DrawCustomMain() is the function that handles displaying the main screen but does not draw the graph on the screen. This function gets called every time the controller updates the display. It is called by ShowInterface that is inside the loop function. This is where you want to handle updating all the data that changes on a frequent or regular basis, ie Temperature, PH, time, relay status, etc.
DrawCustomGraph() is the function that displays the graph on the screen. This function also gets called by ShowInterface but only when it is time to update the graph. The graph gets updated every 12 minutes. It is also called when the menu is exited and you are returning to the main screen. This is where you would want to display the graph or any other custom graphic or text that is not continuously updating (or being updated very often), ie Graph, banner, logo, etc.

Next, we will discuss the contents of DrawCustomMain.

```
ReefAngel.LCD.DrawDate(6, 112);
```

The first thing this function does is draws the date on the display. The date is drawn at the bottom of the screen at position 6,112 (which is 6 pixels from the left, 112 pixels from the top).

```
pingSerial();
```

This called to check if we have any requests on the wifi (if wifi enabled).

```
#if defined DisplayLEDPWM && ! defined RemoveAllLights
     ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params,
ReefAngel.PWM.GetDaylightValue(), ReefAngel.PWM.GetActinicValue());
#else // defined DisplayLEDPWM && ! defined RemoveAllLights
     ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params);
#endif // defined DisplayLEDPWM && ! defined RemoveAllLights
```

This couple of lines is a little complex. There is a check to determine if the controller is able to display the PWM values or not.
The check tests for 2 cases: 1) if DisplayLEDPWM is enabled 2) if RemoveAllLights is NOT enabled
If both cases are true, that means DisplayLEDPWM is enabled and RemoveAllLights is not enabled, then the monitor values are displayed along with the PWM values. Otherwise, the PWM values are not displayed.
If DisplayLEDPWM was disabled (or if RemoveAllLights was enabled) and we did not remove the PWM stuff,

you would receive errors inside Arduino when you tried to upload to the controller. All of this logic is inside the #if #else #endif grouping.

Note: The default Features file has DisplayeLEDPWM enabled and RemoveAllLights disabled, so the PWM values will display by default.

Now that we have explained the logic checks, the actual code to display the monitors is the DrawMonitor functions. If we are displaying the PWM values, this line is used:

```
ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params,
ReefAngel.PWM.GetDaylightValue(), ReefAngel.PWM.GetActinicValue());
```

If we are not displaying PWM values, this line is use:

```
ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params);
```

Both of those lines state that the start of the monitor block will be at 15 pixels from the left and 60 pixels down from the top. It will use the values that are stored in the ReefAngel.Params variable (this contains the T1,T2,T3,PH values, sample usage is shown in the DrawSingleMonitor section later).

```
pingSerial();
```

We check the wifi interface again.

```
byte TempRelay = ReefAngel.Relay.RelayData;
TempRelay &= ReefAngel.Relay.RelayMaskOff;
TempRelay |= ReefAngel.Relay.RelayMaskOn;
ReefAngel.LCD.DrawOutletBox(12, 93, TempRelay);
```

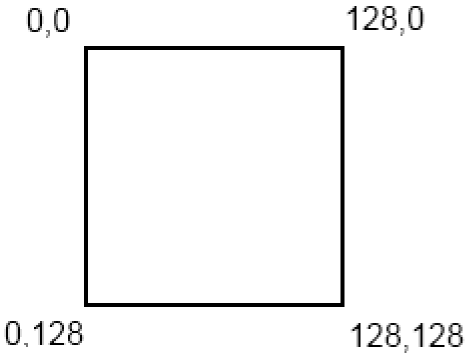
Now we display the outlet box on the screen.

We will discuss the contents of DrawCustomGraph.

```
ReefAngel.LCD.DrawGraph(5,5);
```

There is only one thing that this function does and that is to display the graph. This line says to draw the graph at position 5,5 on the screen (from the top left corner).

Now to discuss more information about the screen and what functions you have to work with.
The screen is a 128x128 square. Everything that is displayed is drawn based on a x-y coordinate system. The top left of the screen is 0,0 and the bottom right is 128,128. The top right is 128,0 and bottom left is 0,128.



The fonts that are displayed on the screen have specific heights and widths. Here's a list of them:

Font	Width x Height (in pixels)	Max Characters (in a row)
Default	5 x 8	21
Font 8x8 and Numbers 8x8	8 x 8	16
Font 8x16 and Numbers 8x16	8 x 16	16
Font 12x16 and Numbers 12x16	12 x 16	8
Numbers 16x16	16 x 16	8

The larger the font, the less text that can be displayed on the screen. The above table lists the fonts that are available for use on the display. In the list, there are Fonts and Numbers. The font means that all the characters are available for use (a - z, A - Z, 0 - 9 and some special characters). The number means that only the numbers are available for use (0 - 9, period and forward slash). The reason for the difference is that the additional fonts require extra storage space. The more fonts to choose from means more space in use and a larger code size. Most of the information being displayed on the main screen is numeric data. So only having numbers means you can get additional sizes without having to add a lot more size to your code.

These additional fonts can be enabled or disabled on the Features tab inside RAGen. Simply check the box next to the font to include and make sure you press the SAVE button to save your changes to your Features file. When you generate a new INO file, the Features file automatically gets generated for you.

Note: You cannot disable/remove the default font. It is used to display the menu and cannot be removed.

Note: When displaying information on the screen, we will be working with the LCD variable inside the ReefAngel class. Since we are doing everything from the INO file, we must always add the "ReefAngel." before the LCD variable so the compiler knows what to do.

Now that we know about the fonts, we have to know how to use them to display the information on the screen. The simplest way to display information is to draw text onto the display. We can accomplish this with using the DrawText function.

DrawText()

ReefAngel.LCD.DrawText(byte fcolor, byte bcolor, byte x, byte y, char *text)

- fcolor – foreground color (see color list index)
- bcolor – background color (see color list index)
- x & y – top left coordinates on screen to start displaying
- text – text to be displayed on screen

There is a color index at the end which displays all the colors that you can choose. You must either choose a named color or specify a hexidecimal value instead. The simplest route is to choose a named color from the chart at the end.

The x,y coordinates indicate the top left corner of where you want the text to be displayed.

This line will display "Hello, World" at the 5th column/pixel from the left and the 10th line/row down:

```
ReefAngel.LCD.DrawText(COLOR_BLACK, COLOR_WHITE, 5, 10, "Hello, World");
```

DrawText will also allow you to display a number instead of a string. This line will display the number 42 in the same spot as "Hello, World" from above:

```
ReefAngel.LCD.DrawText(COLOR_BLACK, COLOR_WHITE, 5, 10, 42);
```

The previous examples all used the default font. If you wanted to use the larger fonts, you would have to use the following functions.

Font	**DrawText Equivalent**
Font_8x8	DrawLargeText(fcolor, bcolor, x, y, text, Font8x8)
Font_8x16	DrawLargeText(fcolor, bcolor, x, y, text, Font8x16)
Numbers_8x8	DrawLargeText(fcolor, bcolor, x, y, text, Num8x8)
Numbers_8x16	DrawLargeText(fcolor, bcolor, x, y, text, Num8x16)
Font_12x16	DrawHugeText(fcolor, bcolor, x, y, text, Font12x16)
Numbers_12x16	DrawHugeText(fcolor, bcolor, x, y, text, Num12x16)
Numbers_16x16	DrawHugeNumbers(fcolor, bcolor, x, y, text)

The arguments are exactly the same except for an extra one allowing you to specify what font you are using. They work exactly the same except there is not a specific function for displaying numbers. You must convert the numbers to strings before displaying. To display Temp1 as a large font, you must do this:

```
void DrawCustomMain()
{
     char text[7];
     ConvertNumToString(text, ReefAngel.Params.Temp1, 10);
     // For PH, you would use this line
     //ConvertNumToString(text, ReefAngel.Params.PH, 100);
     ReefAngel.LCD.DrawLargeText(COLOR_BLACK, COLOR_WHITE, 5, 10, text,
Font8x16);
}
```

Note: The temperatures and PH are all stored as 3 digit numbers. When displayed, we have to add the decimal points in the proper spot otherwise the numbers won't display properly.

Note: ConvertNumToString() is already added to the libraries and ready to use.

This function, ConvertNumToString() is responsible for converting the numbers to a string and adding in the decimal point. The last value used in the function (10 in the example) is for adding in a decimal point. For

temperatures, you will want to use 10 which means to shift the decimal point 1 place to the left. For the PH, you would use 100 (2 places to the left). For all other numbers, use a 1 which means do not shift to the left. If you use a 0, the function will fail.
The rest of the functions that will be discussed only display information using the default font (5x8). If larger fonts are desired to be used with this information, custom functions must be written and handled by yourself.

DrawDate()

DrawDate is a simple function that does just that. It draws the date and time on the display. You must specify the location you want the date and time drawn.

ReefAngel.LCD.DrawDate(x, y)

- x & y – top left coordinates on the screen to place the date

The output is 8 pixels tall and 110 pixels wide.

```
ReefAngel.LCD.DrawDate(6, 112);
```

DrawMonitor()

DrawMonitor is the function that draws the standard grouping of monitors (temperature 1, 2, & 3, PH and PWM percentages). It draws them in a 2 column format and occupies the width of the display. You must specify the location you want it drawn on the screen and provide it the Params structure (contains the temperatures and ph) and the PWM values.

ReefAngel.LCD.DrawMonitor(x, y, Params, Daylight, Actinic)

- x & y – top left coordinates on the screen to place the monitors
- Params – Parameters structure that contains the temperatures and ph
- Daylight – value to display for the daylight percentage
- Actinic – value to display for the actinic percentage

```
ReefAngel.LCD.DrawMonitor(15, 60, ReefAngel.Params,
ReefAngel.PWM.GetDaylightValue(), ReefAngel.PWM.GetActinicValue());
```

This will display the monitors at 15,60 and display the T1,T2,T3,PH,AP & DP values. The output is 30 pixels tall and 110 pixels wide.

DrawOutletBox()

DrawOutletBox is the function that draws the specified outlet box on the screen. You must give it the relay information to be displayed.

ReefAngel.LCD.DrawOutletBox(x, y, RelayData)

- x & y – top left coordinates on the screen to place the outlet box

- RelayData – relay data to be displayed on the screen

The RelayData can be for any of the 9 relays available (1 relay by default and up to 8 additional relays through the expansion modules). In order to display the relay data, the data has to be processed to get the appropriate status of the ports. Here's how you would display the default relay:

```
byte TempRelay = ReefAngel.Relay.RelayData;
TempRelay &= ReefAngel.Relay.RelayMaskOff;
TempRelay |= ReefAngel.Relay.RelayMaskOn;
ReefAngel.LCD.DrawOutletBox(12, 93, TempRelay);
```

And to display the 1st expansion relay:

```
byte TempRelay = ReefAngel.Relay.RelayDataE[0];
TempRelay &= ReefAngel.Relay.RelayMaskOffE[0];
TempRelay |= ReefAngel.Relay.RelayMaskOnE[0];
ReefAngel.LCD.DrawOutletBox(12, 93, TempRelay);
```

As you can see, the only difference is the way you refer to the relay data. The default relay uses the RelayData variable and the expansion relays use the RelayDataE[] array. The array is a 0 based array. This means that the first relay is actually index 0, the second is index 1 and so on with the 8th relay being index 7.

Code Reference	**Expansion Relay**
RelayDataE[0], RelayMaskOnE[0], RelayMaskOffE[0]	Expansion Relay 1
RelayDataE[1], RelayMaskOnE[1], RelayMaskOffE[1]	Expansion Relay 2
RelayDataE[2], RelayMaskOnE[2], RelayMaskOffE[2]	Expansion Relay 3
RelayDataE[3], RelayMaskOnE[3], RelayMaskOffE[3]	Expansion Relay 4
RelayDataE[4], RelayMaskOnE[4], RelayMaskOffE[4]	Expansion Relay 5
RelayDataE[5], RelayMaskOnE[5], RelayMaskOffE[5]	Expansion Relay 6
RelayDataE[6], RelayMaskOnE[6], RelayMaskOffE[6]	Expansion Relay 7
RelayDataE[7], RelayMaskOnE[7], RelayMaskOffE[7]	Expansion Relay 8

The output on the screen is 12 pixels tall and 104 wide.

DrawSingleMonitor()

DrawSingleMonitor is a function that displays either the T1, T2, T3, PH, AP or DP values on the screen. This function can display any numeric value. It handles clearing the section of the screen to display and also adding in the decimal point for you. It does not print any text label in front of the value. You must do that yourself. It get's called by DrawMonitor. You can use this function if you only want to display specific values or if you want the values arranged in a different order/position than the default.

ReefAngel.LCD.DrawSingleMonitor(Temp, fcolor, x, y, decimal)

- Temp – value to be displayed
- fcolor – color of the text (background color is the default background color)
- x, y – top left coordinates on the screen to place the text
- decimal – where the decimal is to be placed. 1 – 0 digit/do not move, 10 – 1 digit, 100 – 2 digits. Ex. 755 is the value with 10 as the decimal. The display is **75.5**. 924 is the value with 100 as the decimal. The display is **9.24**. 90 is the value with 1 as the decimal. The display is **90**.

Note: If you are displaying temperatures, then decimal should be 10. If you are displaying PH, then decimal should be 100. If you are displaying the AP or DP (PWM values), then decimal should be 1. A decimal value of 0 will cause the function to fail.

Here are examples for all the values that are displayed inside DrawMonitor:

```
// Display the T1 temperature at 5,5
ReefAngel.LCD.DrawSingleMonitor(ReefAngel.Params.Temp1, COLOR_BLACK, 5, 5,
10);
// Display the T2 temperature at 5,5
ReefAngel.LCD.DrawSingleMonitor(ReefAngel.Params.Temp2, COLOR_BLACK, 5, 5,
10);
// Display the T3 temperature at 5,5
ReefAngel.LCD.DrawSingleMonitor(ReefAngel.Params.Temp3, COLOR_BLACK, 5, 5,
10);
// Display the PH at 20,5
ReefAngel.LCD.DrawSingleMonitor(ReefAngel.Params.PH, COLOR_BLACK, 20, 5,
100);
// Display the Actinic PWM channel at 30,5
ReefAngel.LCD.DrawSingleMonitor(ReefAngel.PWM.GetActinicValue(),
COLOR_BLACK, 30, 5, 1);
// Display the Daylight PWM channel at 30,5
ReefAngel.LCD.DrawSingleMonitor(ReefAngel.PWM.GetDaylightValue(),
COLOR_BLACK, 30, 5, 1);
```

The output on the screen is 8 pixels tall and 30 pixels wide.

pingSerial()

pingSerial is the function that needs to be called to process any requests that come in on the wifi interface. If you do not have wifi enabled, this function does nothing. If wifi is enabled, it processes any wifi related requests.

```
pingSerial();
```

Clear()

Clear is the function that clears the screen. You need to give it a color to "clear" the screen with and a specified area. The area is rectangular. You must specify the top left corner and bottom right corner. The area inside the rectangle is filled in with the given color.

ReefAngel.LCD.Clear(color, x1, y1, x2, y2)

- color to fill in rectangle (see color list index)
- x1,y1 - top left corner
- x2,y2 - bottom right corner

Using this function is beneficial when changing values being displayed on the screen. Here's an example of why you would use this. The PWM percentage can be anywhere from 0 to 100 percent. This would be anywhere from 1 to 3 digits being displayed. If you display the percent at the same spot, you will want to make sure you clear out the entire area that the value could be displayed before you display the new value. If you do not, you will get extra, unwanted values on the screen.

Say the PWM value is 25%. That value is displayed on the screen starting at 5,5. Now you change the value to 9%. That value will also be displayed at 5,5. If you do not clear out the entire space that the value will be at, only the 2 from 25% will be replaced. When it is replaced, the resulting display will be 95% instead of 9%.

```
// Clear out a box from 5,10 to 35,18 with the color white
ReefAngel.LCD.Clear(COLOR_WHITE, 5, 10, 35, 18);
```

DrawGraph()

DrawGraph is the function that displays the graph of all 3 temperature probes and the PH probe. It draws the graph at the given coordinates.

ReefAngel.LCD.DrawGraph(x, y)

- x,y – top left coordinates on the screen to draw the graph

The coordinates are the top left corner of the graph to be displayed. The output on the screen is 50 pixels tall and 121 pixels wide.

```
ReefAngel.LCD.DrawGraph(5,5);
```

Here are a list of the default values that can be used if you desire. These values are located in the **ReefAngel_Globals.h** file. They can be used anywhere that a color (fcolor or bcolor) are used. The values are in bold below.

```
// Text color for the T1 temp probe (homescreen)
#define T1TempColor          COLOR_RED
// Text color for the T2 temp probe (homescreen)
#define T2TempColor          COLOR_CHOCOLATE
// Text color for the T3 temp probe (homescreen)
#define T3TempColor          COLOR_MEDIUMORCHID
// Text color for the PH probe (homescreen)
#define PHColor              COLOR_SEAGREEN
// Text color for Daylight PWM value (homescreen)
#define DPColor              COLOR_SADDLEBROWN
// Text color for the Actinic PWM value (homescreen)
#define APColor              COLOR_DARKSLATEBLUE
```

```
// color of the PH calibration text values
#define CalibrateColor        COLOR_RED
// color of the text on Feeding Mode, Water Change Mode and
// Version screens
#define ModeScreenColor       COLOR_BLUE
// color of the date on home screen
#define DateTextColor         COLOR_RED
// Border around the outlet box on home screen
#define OutletBorderColor     0x54
// Outlet box background color when outlet is on (homescreen)
#define OutletOnBGColor       COLOR_GREEN
// Outlet box text (number) color when outlet is on (homescreen)
#define OutletOnFGColor       COLOR_WHITE
// Outlet box background color when outlet is off (homescreen)
#define OutletOffBGColor      COLOR_RED
// Outlet box text (number) color when outlet is off (homescreen)
#define OutletOffFGColor      COLOR_WHITE
// OK & Cancel button border color (setup screens)
#define BtnBorderColor        COLOR_BLACK
// OK & Cancel button ACTIVE (selected) background color (setup screens)
#define BtnActiveColor        COLOR_GRAY
// OK & Cancel button INACTIVE (not selected)
// background color (setup screens)
#define BtnInactiveColor      COLOR_LIGHTGRAY
// Background color for selected items (Menu & Setup screens)
#define SelectionBGColor      COLOR_BLUE
// Foreground (text) color for selected items (Menu & Setup screens)
#define SelectionFGColor      COLOR_WHITE
// Default background color
#define DefaultBGColor        COLOR_WHITE
// Default text color
#define DefaultFGColor        COLOR_BLACK
// color of the dotted line in the middle of the graph
#define GraphDotLineColor     0x49
```

Examples

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

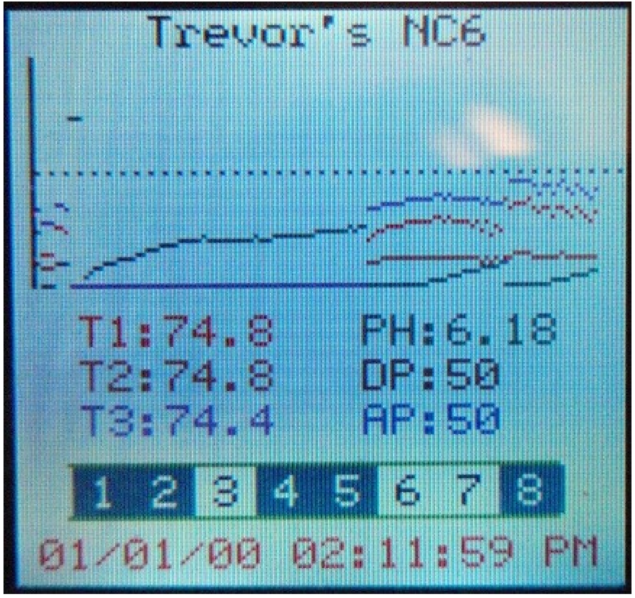
Banner

This example mimics the default main screen but adds a banner above the graph.

```
void DrawCustomMain()
{
     ReefAngel.LCD.DrawDate(6, 118);
     pingSerial();
     ReefAngel.LCD.DrawMonitor(15, 68, ReefAngel.Params,
               ReefAngel.PWM.GetDaylightValue(),
               ReefAngel.PWM.GetActinicValue());
     pingSerial();
     byte TempRelay = ReefAngel.Relay.RelayData;
     TempRelay &= ReefAngel.Relay.RelayMaskOff;
     TempRelay |= ReefAngel.Relay.RelayMaskOn;
     ReefAngel.LCD.DrawOutletBox(12, 101, TempRelay);
}

void DrawCustomGraph()
{
     // Change the 30 to adjust the horizontal position of the text
     ReefAngel.LCD.DrawText(DefaultFGColor, DefaultBGColor, 30, 2,
"Trevor's NC6");
     ReefAngel.LCD.DrawGraph(5, 11);
}
```

Note: The banner can be a maximum of 21 characters with the standard (5x8) font.

If you want to center the text on the controller, you need to take the length of your banner and multiply that by 5 (assuming you are using the standard font which is 5 pixels wide). Once you get that value, then you need to subtract it from 128 and divide that in half. You divide it by 2 so you have equal space on both sides. Using the above example: “Trevor's NC6” is 12 characters wide. 12 x 5 = 60 pixels wide. Subtract that width from 128 pixels and you get 68 (128 – 60). Now, you need to divide 68 by 2, for equal space on both sides. That yields 34. I decided to shift it left a little. I chose 30. You can adjust to whatever looks good to you.



```
Trevor's NC6

T1:74.8    PH:6.18
T2:74.8    DP:50
T3:74.4    AP:50
1 2 3 4 5 6 7 8
01/01/00 02:11:59 PM
```

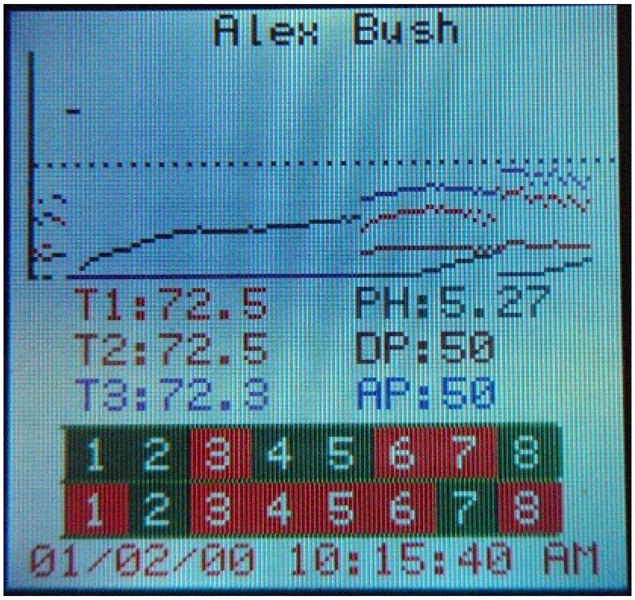
Banner with 2 Relays

This will show a banner and also display 2 relay boxes with the default main screen.

```
void DrawCustomMain()
{
  ReefAngel.LCD.DrawDate(6, 119);
  pingSerial();
  ReefAngel.LCD.DrawMonitor(15, 63, ReefAngel.Params,
                            ReefAngel.PWM.GetDaylightValue(),
                            ReefAngel.PWM.GetActinicValue());
  pingSerial();
  // draw main relay
  byte TempRelay = ReefAngel.Relay.RelayData;
  TempRelay &= ReefAngel.Relay.RelayMaskOff;
  TempRelay |= ReefAngel.Relay.RelayMaskOn;
  ReefAngel.LCD.DrawOutletBox(12, 93, TempRelay);
#ifdef RelayExp
  // draw 1st expansion relay
  TempRelay = ReefAngel.Relay.RelayDataE[0];
  TempRelay &= ReefAngel.Relay.RelayMaskOffE[0];
  TempRelay |= ReefAngel.Relay.RelayMaskOnE[0];
  ReefAngel.LCD.DrawOutletBox(12, 105, TempRelay);
#endif  // RelayExp
}

void DrawCustomGraph()
{
  // Change the 45 to adjust the horizontal position of the text
  ReefAngel.LCD.DrawText(DefaultFGColor, DefaultBGColor, 45, 2, "Alex
Bush");
  ReefAngel.LCD.DrawGraph(5, 10);
}
```

The above code segment makes sure that RelayExp (relay expansion) code is enabled. If it is not enabled and you try to reference that section of code, you will get errors inside Arduino and your code will not compile. If you know you have it enabled you don't need to have the #ifdef and #endif lines, but for an example with others it's best to add this functionality. It will not affect the code at all.

This example uses the same logic as with the previous example with the Banner. The only difference is that it now references an additional relay box (the first expansion relay).



```
Alex Bush
T1:72.5    PH:5.27
T2:72.5    DP:50
T3:72.3    AP:50
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8
01/02/00 10:15:40 AM
```

Large fonts

This example uses 2 large fonts: 16x16 and 8x16. You must enable the fonts in your features file.

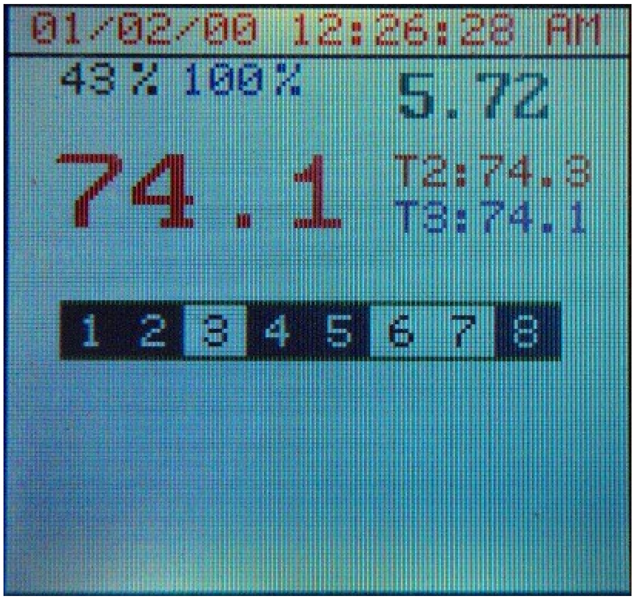
```
#define NUMBERS_8x16
#define NUMBERS_16x16
```

Note: This will use more code space. You may want to enable SIMPLE_MENU to help reduce the code size.

We have to create an additional function that converts numbers to a string for use with the text functions. We covered this earlier in the guide, but here it is in action.

```
void DrawCustomMain()
{
  byte x = 6;
  byte y = 2;
  byte t;
  ReefAngel.LCD.DrawDate(6, 2);
  ReefAngel.LCD.Clear(COLOR_BLACK, 1, 11, 132, 11);
  pingSerial();

  x = 12;
  y += MENU_START_ROW+1;
  ReefAngel.LCD.DrawSingleMonitor(ReefAngel.PWM.GetDaylightValue(),
DPColor, x, y, 1);
  t = intlength(ReefAngel.PWM.GetDaylightValue()) + 1;
  t *= 5;
  ReefAngel.LCD.DrawText(DPColor, DefaultBGColor, x+t, y, "%");
  t += 10;
  x += t;
  ReefAngel.LCD.DrawSingleMonitor(ReefAngel.PWM.GetActinicValue(),
APColor, x, y, 1);
  t = intlength(ReefAngel.PWM.GetActinicValue())  + 1;
  t *= 5;
  ReefAngel.LCD.DrawText(APColor, DefaultBGColor, x+t, y, "%");
  t += 10;
  x += t;
  char text[7];
  ConvertNumToString(text, ReefAngel.Params.PH, 100);
  ReefAngel.LCD.Clear(DefaultBGColor, x+16, y, x+45, y+16);
  ReefAngel.LCD.DrawLargeText(PHColor, DefaultBGColor, x+16, y, text, 
Num8x16);
  pingSerial();

  ConvertNumToString(text, ReefAngel.Params.Temp1, 10);
  y += MENU_START_ROW*2;
  x = 10;
  ReefAngel.LCD.Clear(DefaultBGColor,x,y,x+(16*4),y+16);
  pingSerial();
  ReefAngel.LCD.DrawHugeNumbers(T1TempColor, DefaultBGColor, x, y, text);
  pingSerial();
  x += (16*4) + 8;
  ReefAngel.LCD.DrawText(T2TempColor,DefaultBGColor,x,y,"T2:");
  ReefAngel.LCD.DrawSingleMonitor(ReefAngel.Params.Temp2, T2TempColor, 
x+18, y, 10);
  ReefAngel.LCD.DrawText(T3TempColor,DefaultBGColor,x,y+10,"T3:");
  ReefAngel.LCD.DrawSingleMonitor(ReefAngel.Params.Temp3, T3TempColor, 
x+18, y+10, 10);

  y += 16*2;
  x = 12;
  byte TempRelay = ReefAngel.Relay.RelayData;
  TempRelay &= ReefAngel.Relay.RelayMaskOff;
  TempRelay |= ReefAngel.Relay.RelayMaskOn;
  ReefAngel.LCD.DrawOutletBox(x, y, TempRelay);
}

void DrawCustomGraph()
{
}
```

You will also note that there is not a graph being drawn. So this function can be left empty. There is a lot of code listed above. Most of the code you have seen before but some may be new to you. The function CovertNumToString adds in the decimal place at the appropriate spot for temperatures and PH.



Final Note

One final note, experiment with the functions and code. It does appear to be challenging and confusing but when you plan things out, it is pretty straight forward. I have discussed all the functions that you will need to create your own custom main screen. These are the same functions that I have used in creating the examples. You cannot damage the controller with the code. The worst thing that can happen is that the code will not compile or you display values/text that does not fit on the screen.

A suggestion would be to design a layout on paper of how/what you want your screen to look like and then work at mimicking it. If you get stuck, ask questions in the forums (http://forum.reefangel.com/).

Lastly, you are encouraged to share your own custom main screen with everybody. Who knows, somebody may like your screen and want to use it. If you share, please post a screenshot of your main screen along with the code used to create it (DrawCustomMain and DrawCustomGraph functions) in the forums (http://forum.reefangel.com/).

Happy Coding!

curt

Color Chart

Name	dec	#	swatch	Name	dec	#	swatch
BLACK	0	00	⬛	YELLOWGREEN	153	99	🟩
NAVY	2	02	🟦	SKYBLUE	155	9B	🟦
BLUE	3	03	🟦	LIGHTGREEN	158	9E	🟩
DARKGREEN	12	0C	🟩	BROWN	164	A4	🟥
GREEN	16	10	🟩	MEDIUMVIOLETRED	166	A6	🟪
TEAL	18	12	🟦	MEDIUMORCHID	170	AA	🟪
LIGHT_BLUE	19	13	🟦	DARKGOLDENROD	176	B0	🟫
DEEPSKYBLUE	23	17	🟦	ROSYBROWN	178	B2	🟫
DARKTURQUOISE	26	1A	🟦	DARKKHAKI	181	B5	🟨
LIME	28	1C	🟩	SILVER	182	B6	⬜
SPRINGGREEN	29	1D	🟩	LIGHTSTEELBLUE	183	B7	🟦
MEDIUMSPRINGGREEN	30	1E	🟩	POWDERBLUE	187	BB	🟦
CYAN	31	1F	🟦	GREENYELLOW	189	BD	🟩
MIDNIGHTBLUE	37	25	🟦	PALETURQUOISE	191	BF	🟦
DARKSLATEGREY	41	29	⬛	CRIMSON	197	C5	🟥
FORESTGREEN	48	30	🟩	CHOCOLATE	204	CC	🟧
SEAGREEN	49	31	🟩	INDIANRED	205	CD	🟥
DODGERBLUE	51	33	🟦	PALEVIOLETRED	206	CE	🟥
LIGHTSEAGREEN	54	36	🟩	ORCHID	207	CF	🟪
LIMEGREEN	57	39	🟩	DARKSALMON	209	D1	🟧
INDIGO	66	42	🟪	PLUM	211	D3	🟪
DARKSLATEBLUE	74	4A	🟪	GOLDENROD	212	D4	🟨
DARKOLIVEGREEN	77	4D	🟩	TAN	214	D6	🟫
ROYALBLUE	79	4F	🟦	THISTLE	215	D7	🟪
STEELBLUE	82	52	🟦	LIGHTGRAY	218	DA	⬜
MEDIUMSEAGREEN	85	55	🟩	LAVENDER	219	DB	⬜
TURQUOISE	90	5A	🟦	LIGHTCYAN	223	DF	⬜
SLATEBLUE	106	6A	🟪	RED	224	E0	🟥
DIMGREY	109	6D	⬛	MAGENTA	227	E3	🟪
MEDIUMSLATEBLUE	111	6F	🟪	DEEPPINK	230	E6	🟥
OLIVEDRAB	112	70	🟩	ORANGERED	232	E8	🟧
SLATEGREY	114	72	⬛	TOMATO	237	ED	🟥
CORNFLOWERBLUE	115	73	🟦	HOTPINK	238	EE	🟪
MEDIUMAQUAMARINE	122	7A	🟩	DARKORANGE	240	F0	🟧
LAWNGREEN	124	7C	🟩	SALMON	241	F1	🟧
AQUAMARINE	126	7E	🟩	LIGHTCORAL	242	F2	🟥
MAROON	128	80	🟥	VIOLET	243	F3	🟪
PURPLE	130	82	🟪	ORANGE	244	F4	🟧
DARKORCHID	134	86	🟪	SANDYBROWN	245	F5	🟧
BLUEVIOLET	135	87	🟪	PINK	246	F6	🟥
SADDLEBROWN	136	88	🟫	GOLD	248	F8	🟨
SIENNA	137	89	🟫	KHAKI	250	FA	🟨
MEDIUMPURPLE	143	8F	🟪	ANTIQUEWHITE	251	FB	⬜
OLIVE	144	90	🟩	YELLOW	252	FC	🟨
GRAY	146	92	⬛	LIGHTGOLDENRODYELLOW	254	FE	⬜
DARKSEAGREEN	150	96	🟩	WHITE	255	FF	⬜