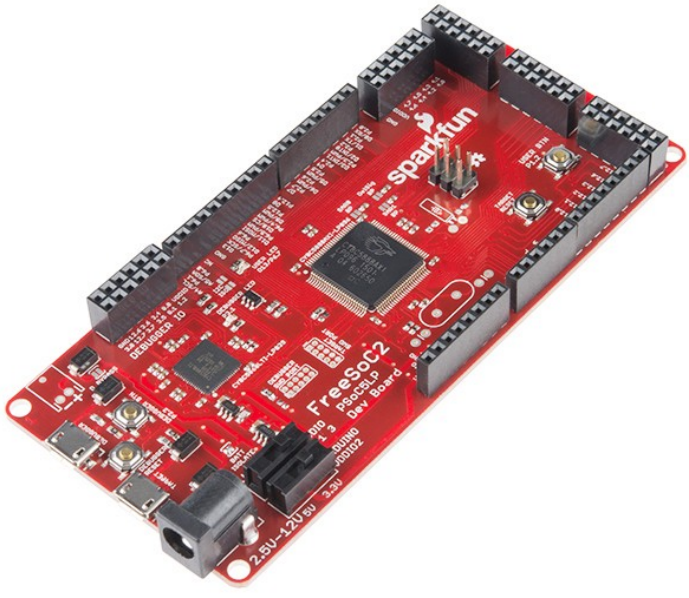
TENET TECHNETRONICS

2016

Floating Point Values in FreeSoc2 (PSoC 5LP) using PSoC Creator



Author: Gurudatta Palankar

Version: 1.0

Introduction:

The FreeSoC2 micro-controller based on the PSoC 5LP (Programmable System on a Chip) brings together features of the programmable devices and micro-controller-type systems on chips into one package. By placing a programmable fabric between the peripherals and the pins, the FreeSoC2 allows any function to be routed to any pin! Moreover, the on-board PSoC includes a number of programmable blocks which allow the user to define arbitrary digital and analog circuits for their specific application. To get the most out of the device, you will need to use the PSoC Creator IDE.

Step 1: Open PSOC creator IDE.

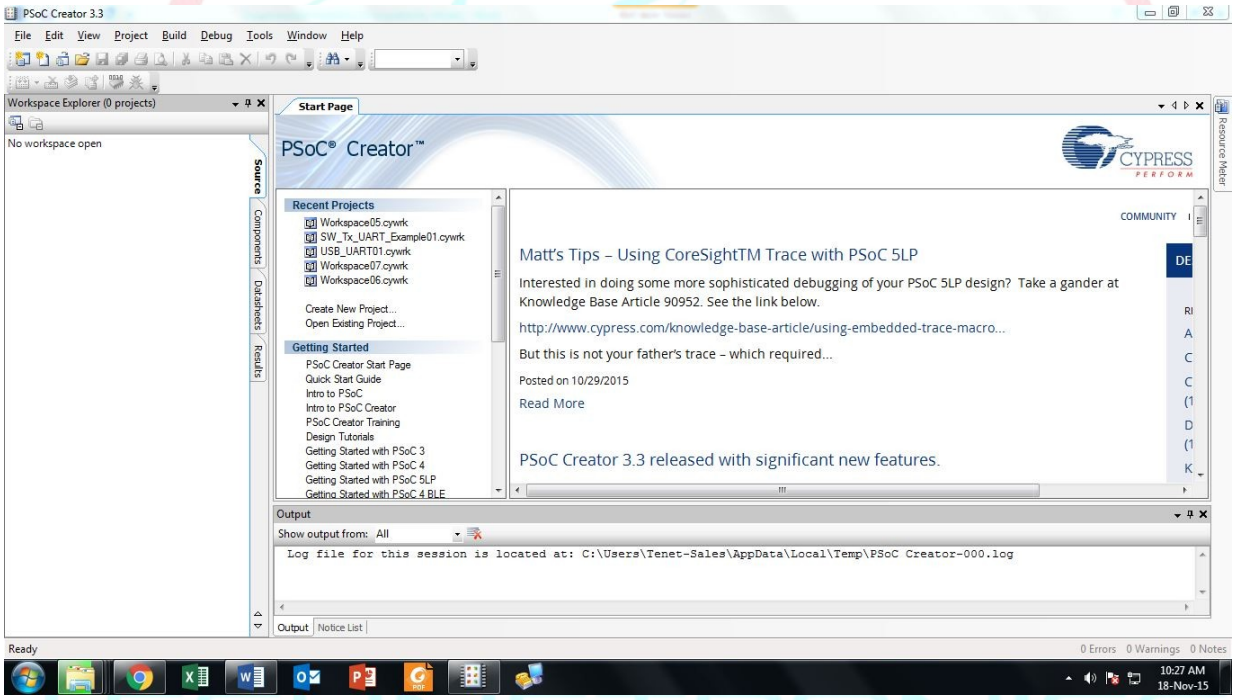


Figure 1

Step 2: File-> new project -> design -> PSoC 5LP design & save with desired name.

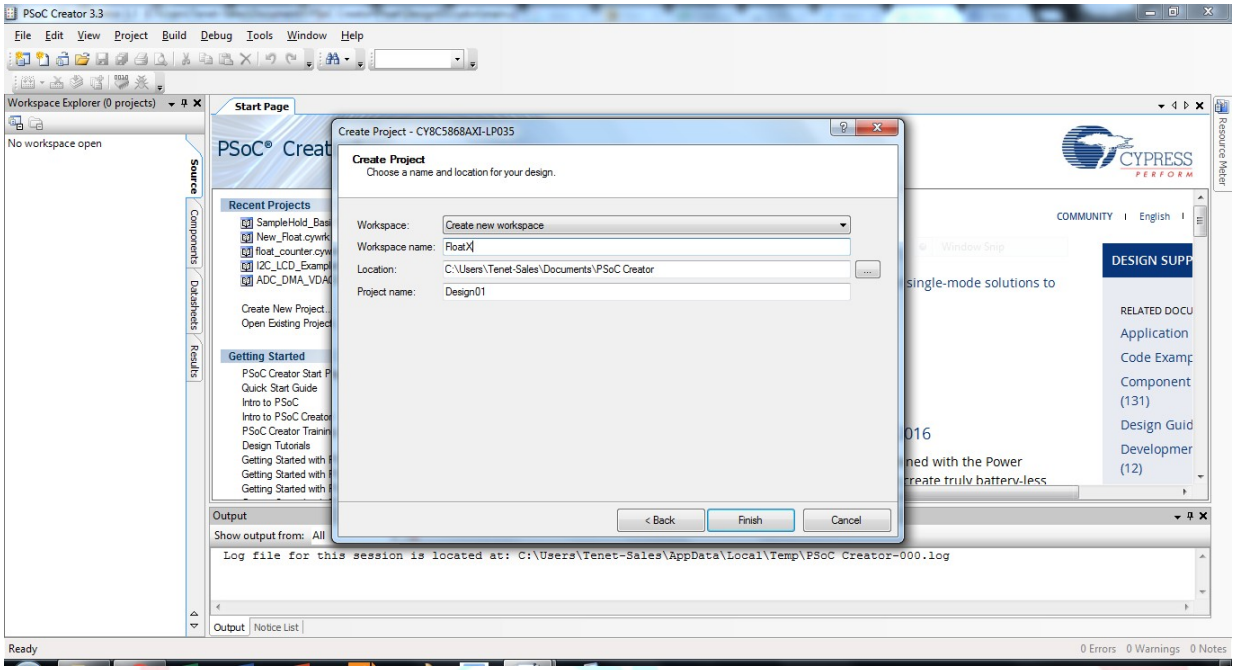


Figure 1

Step 3: Open TopDesign.cysch from workspace explorer.

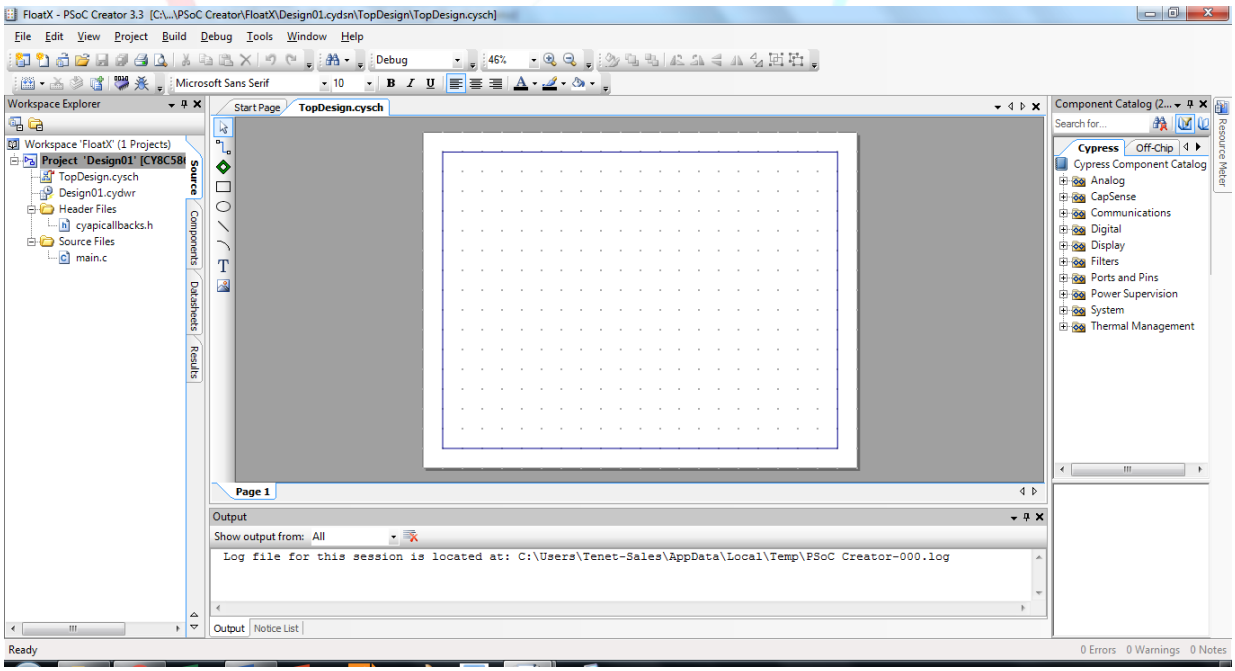


Figure 2

Step 4: Search for UART component from the Component catalog on right side of the window. Drag the UART component onto the workspace. Double click on the UART component and change the name if you wish to. Configure the UART as 8-bit and baud rate of 9600 bps.

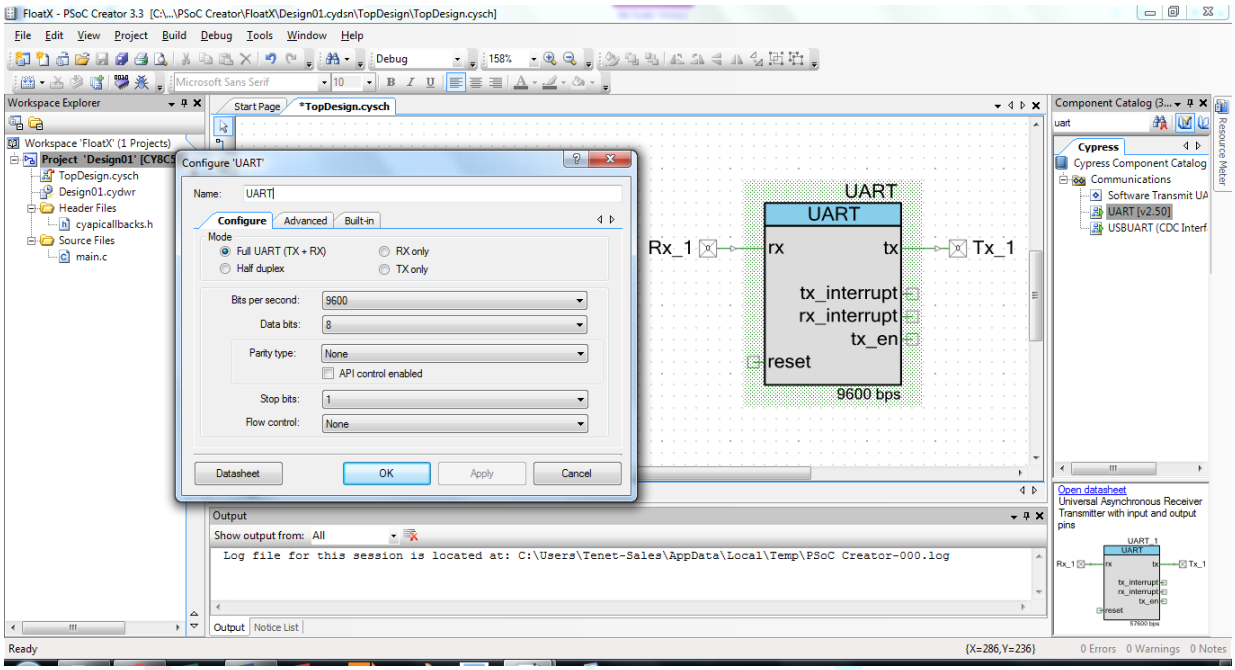


Figure 3

Step 5: After configuring build the project. As we can generate user-defined APIs which will ease us while writing code. We can see APIs generated in the Workspace Explorer on the left side of the window.

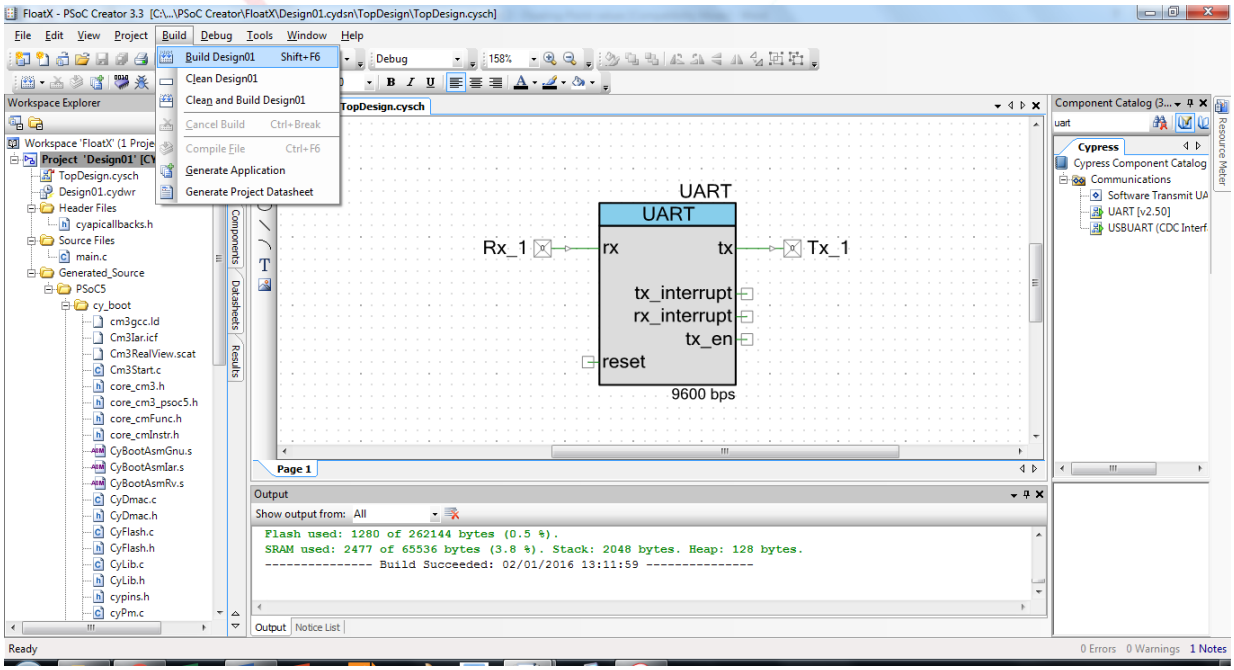


Figure 5

Step 6: Click on main.c from Workspace Explorer. Write the code and Build it.

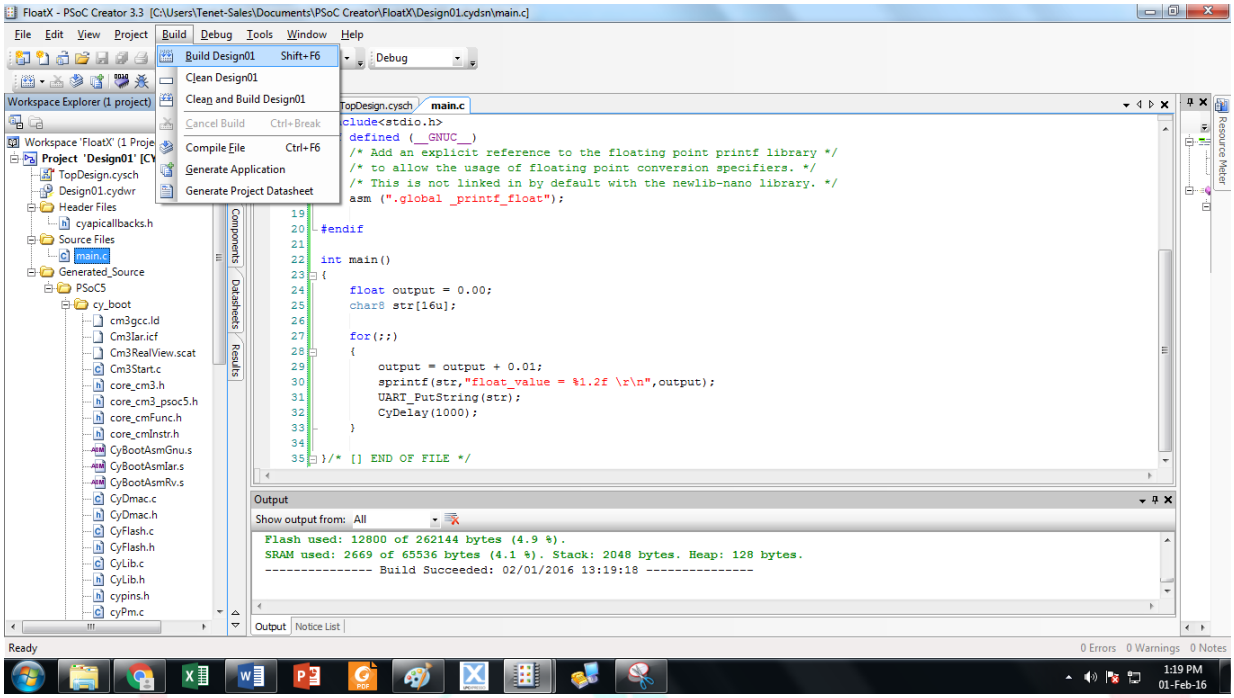


Figure 6

CODE:

```
#include<stdio.h>
#if defined (__GNUC__)
    /* Add an explicit reference to the floating point printf library */
    /* to allow the usage of floating point conversion specifiers. */
    /* This is not linked in by default with the newlib-nano library. */
    asm (".global _printf_float");

#endif

int main()
{
    float output = 0.00;
    char8 str[16u];

    for(;;)
    {
        output = output + 0.01;
        sprintf(str,"value = %1.2f \r\n",output);
        UART_PutString(str);
        CyDelay(1000);
    }

}
```

Step 7: Right click on Project and click on Build Settings...

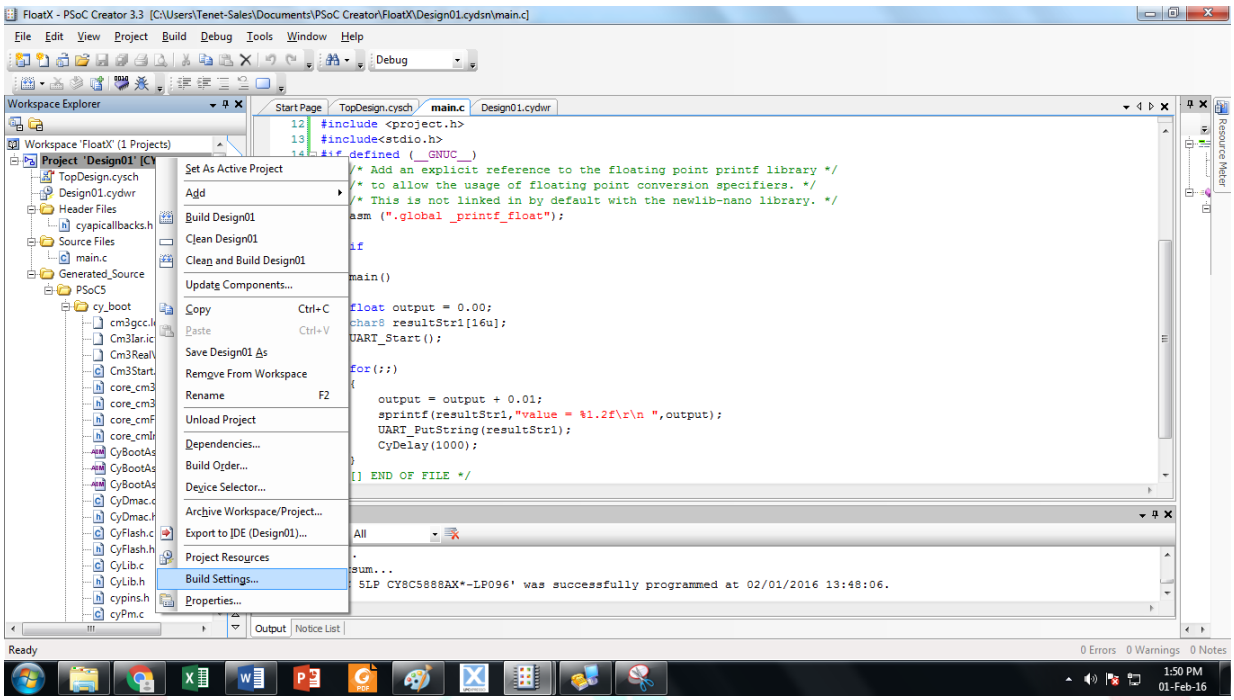


Figure 7

Step 8: Go to Compiler >> Optimization, select Optimization Level as None.

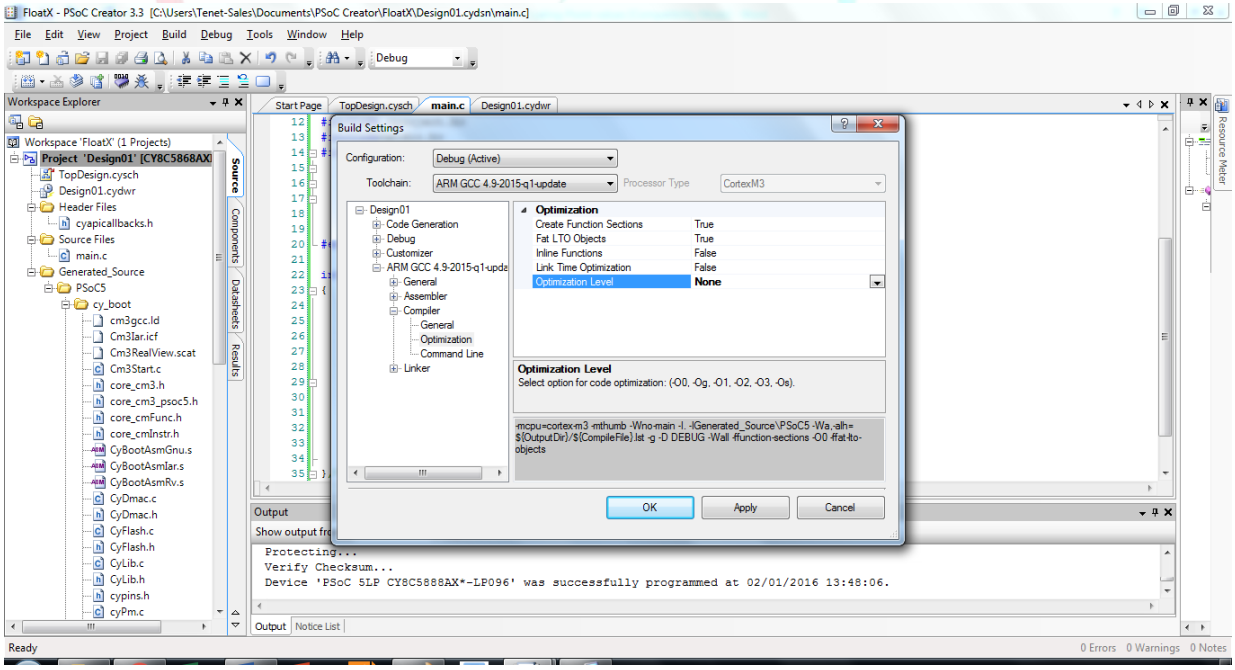


Figure 8

Step 9: Double click on Design01.cydwr

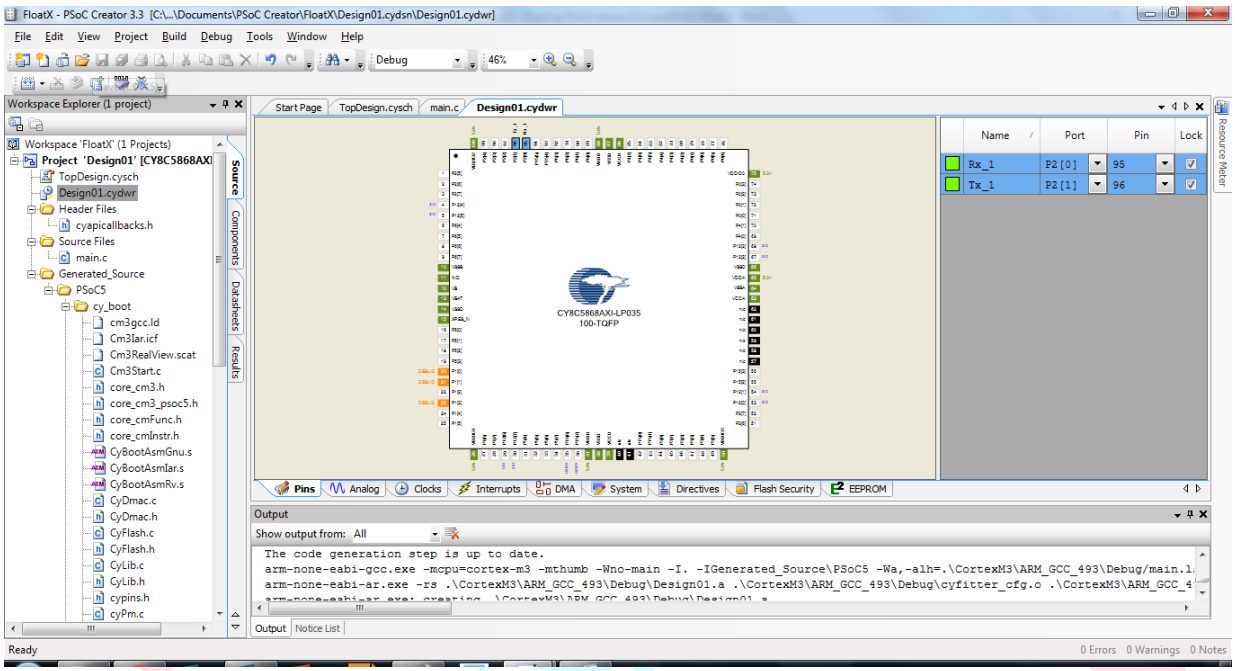


Figure 9

Step 10: Click on System on Design01.cydwr and increase Heap Size and Stack Size.

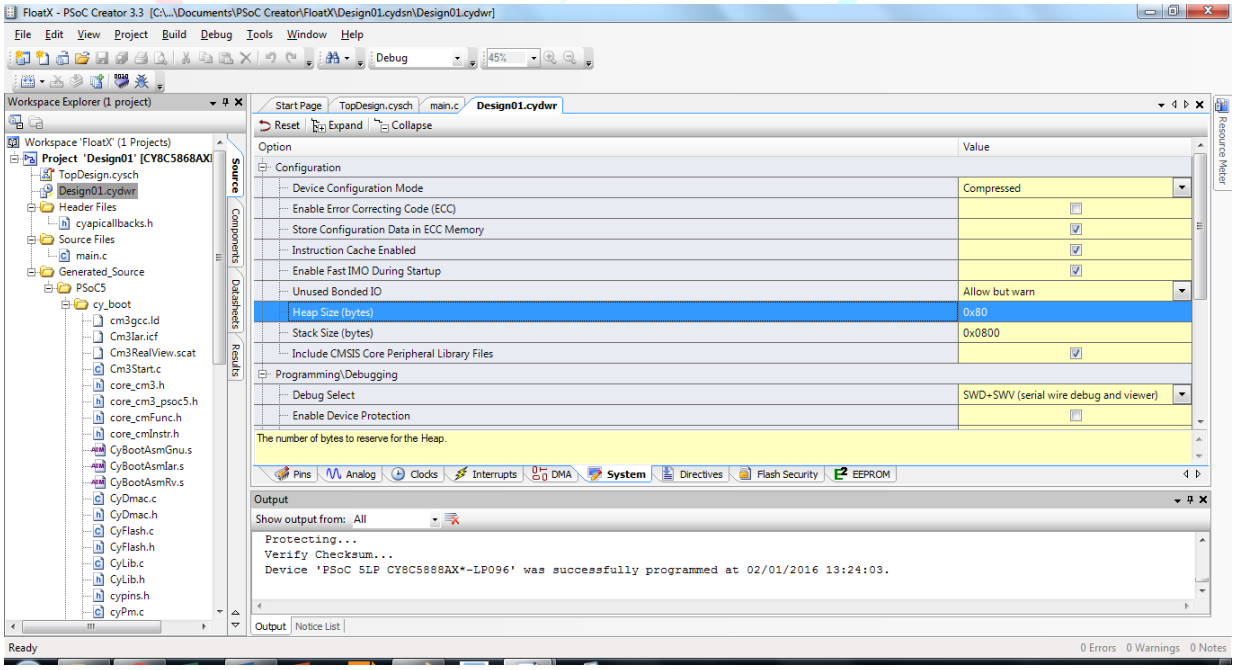


Figure 10

Step 11: Here we have increased Heap Size for 0x1000 and Stack Size for 0x4000.

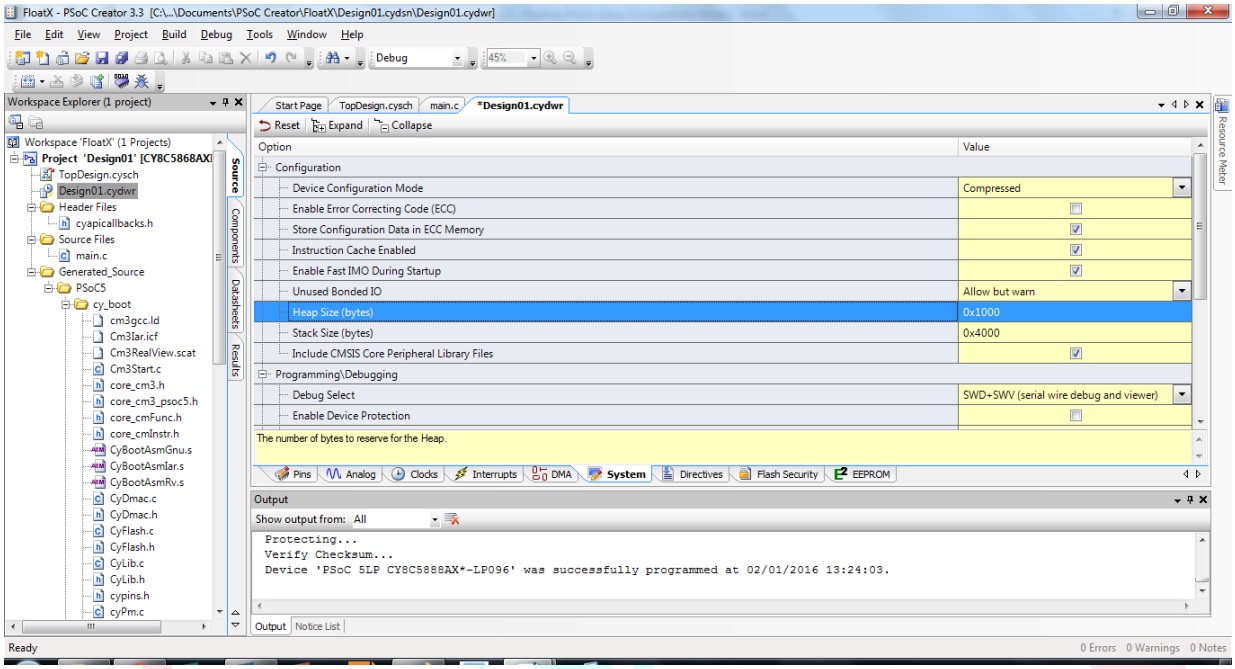


Figure 11

Step 12: Finally, assign pins to desired port and build it.

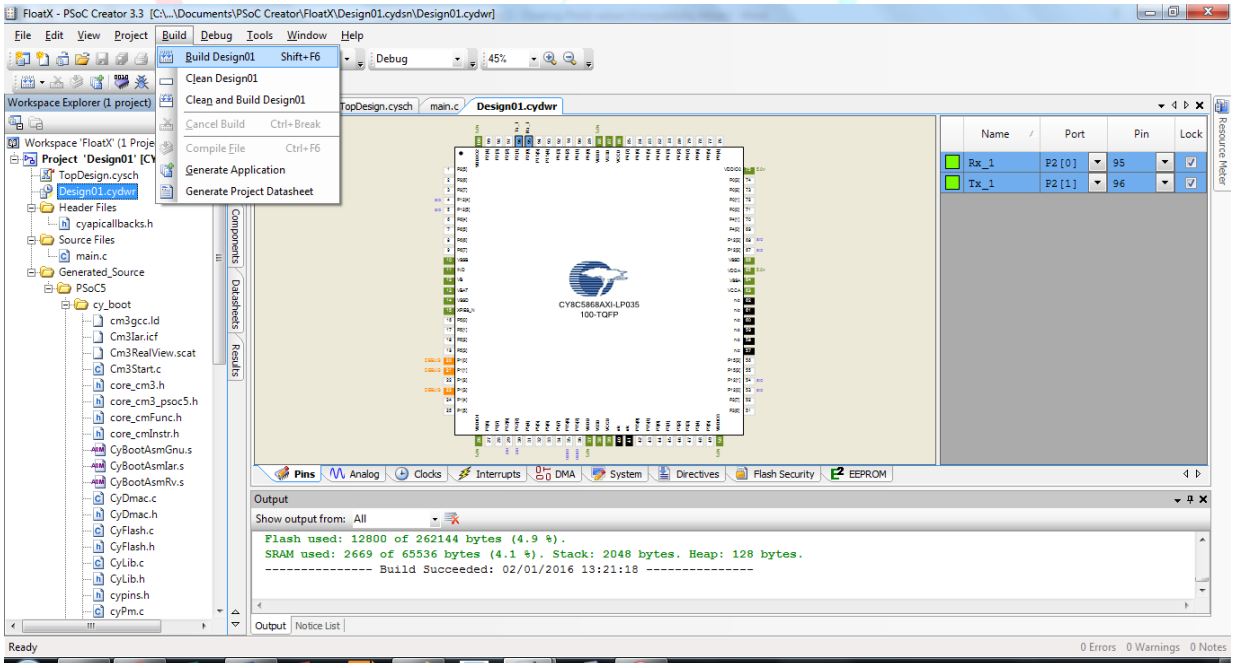


Figure 12

Step 13: If all goes well, go to Debug and click on Program.

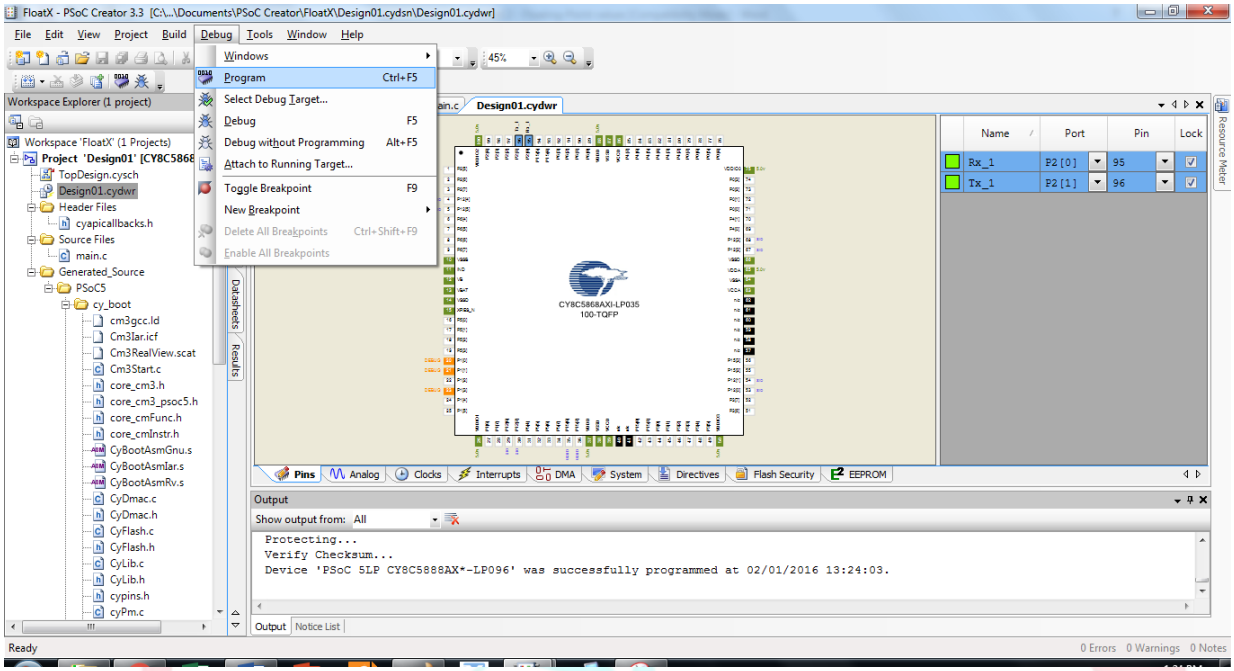


Figure 13

Step 14: Now to display floating point numbers onto screen we need a terminal. A terminal used to display via COM ports. You can use any terminal such as Putty, Hyper-Terminal, X-CTU and so on. Here we have used Hyper-Terminal. Select appropriate COM port.

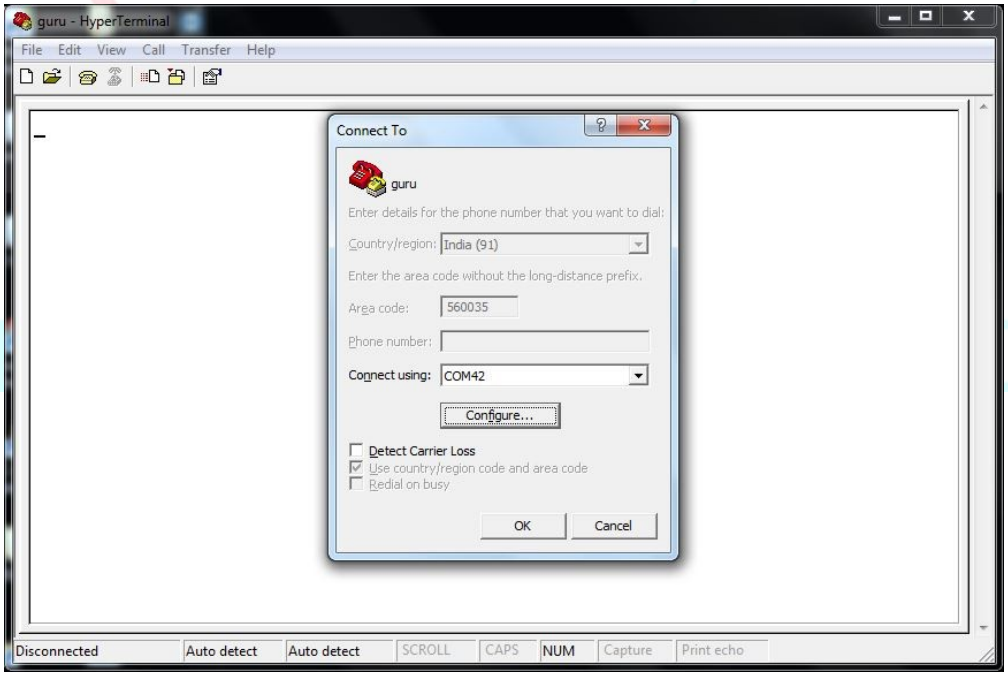


Figure 14

Step 15: Configure as 8-bit and baud rate of 9600 bps.

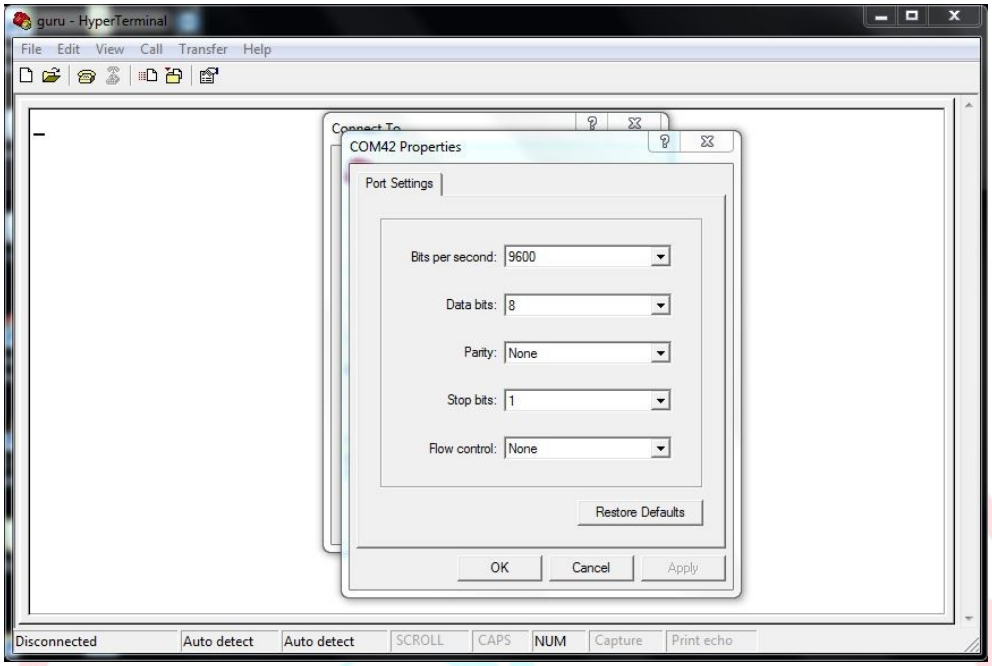


Figure 15

Step 16: Now we can see floating point values on the terminal.

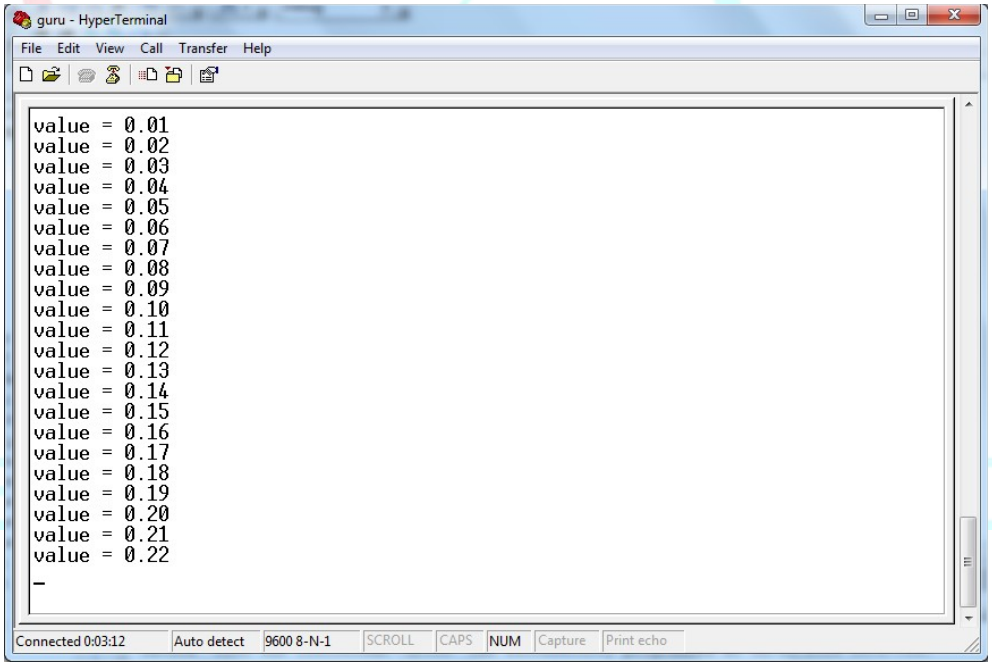


Figure 16

CIRCUIT EXPLAINATION:

The UART module we just added is typically used to achieve a serial communication link using some of the GPIO pins. In this case we want to be able to transfer our data through the USB interface to our PC to avoid the need of any additional cables and equipment.

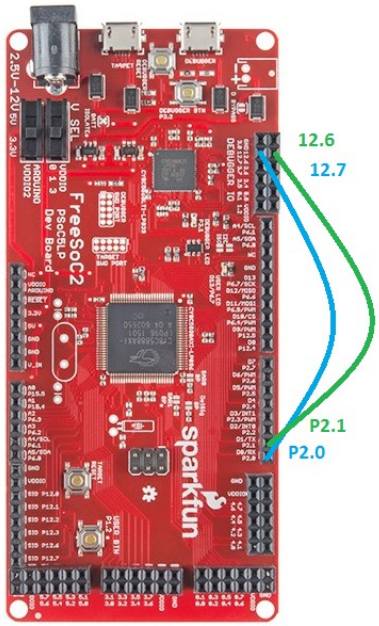


Figure 17

Pin P2.0 is RX and P2.1 is TX. We connect the two pins to RX/TX pins on the debugger/programmer which is 12.6 (RX) and 12.7 (TX). Note that the RX pin from the PSoC 5LP MCU should be connected to the TX pin of the debugger. And the P2.1 TX pin should be connected to the debugger's RX pin (12.6).

OUTPUT:

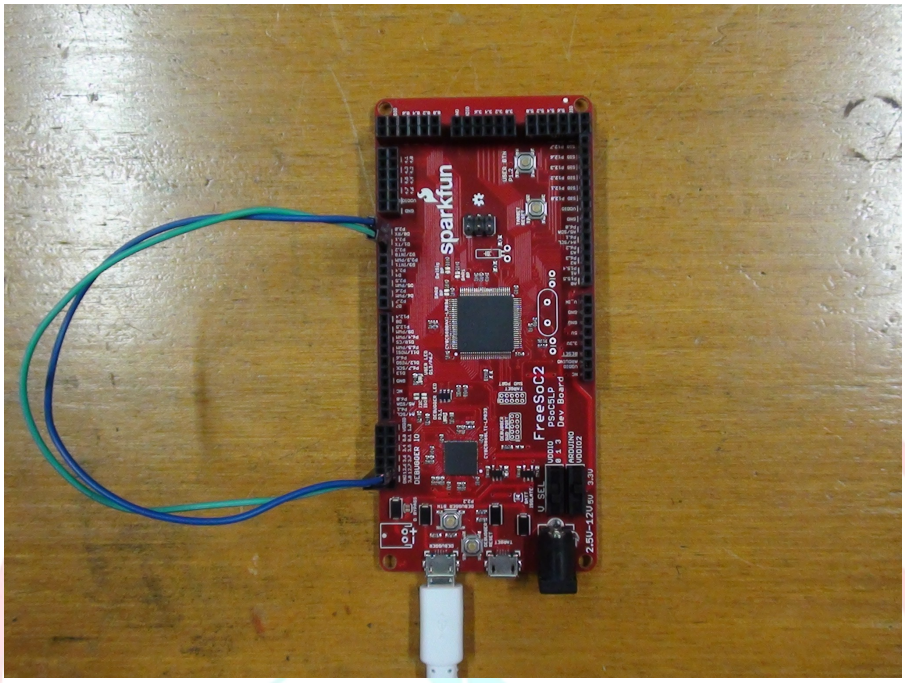


Figure 18

For product link:

1. **http://www.tenettech.com/product/7241/freesoc2-development-board-psoc5lp**
2. **http://www.tenettech.com/product/2442/16-x-2-character-lcd-display-with-backlight-jhd162a-green**

For more information please visit: www.tenettech.com

For technical query please send an e-mail: info@tenettech.com