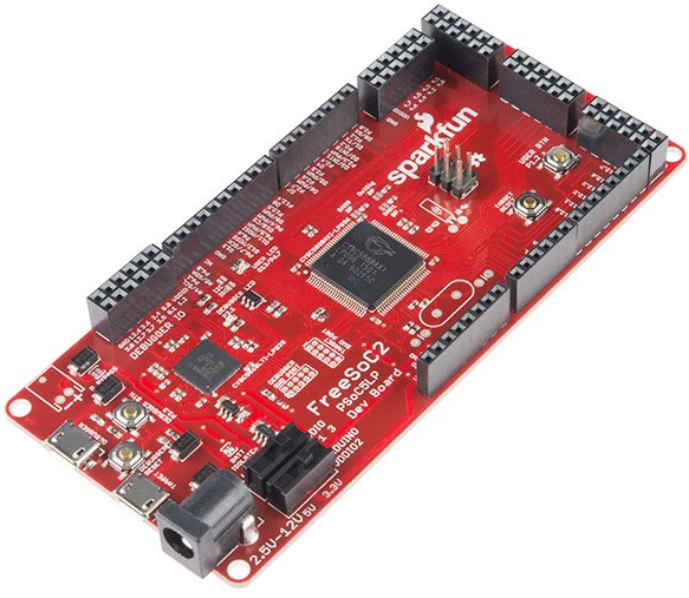
TENET TECHNETRONICS

2016

Software Serial with 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.

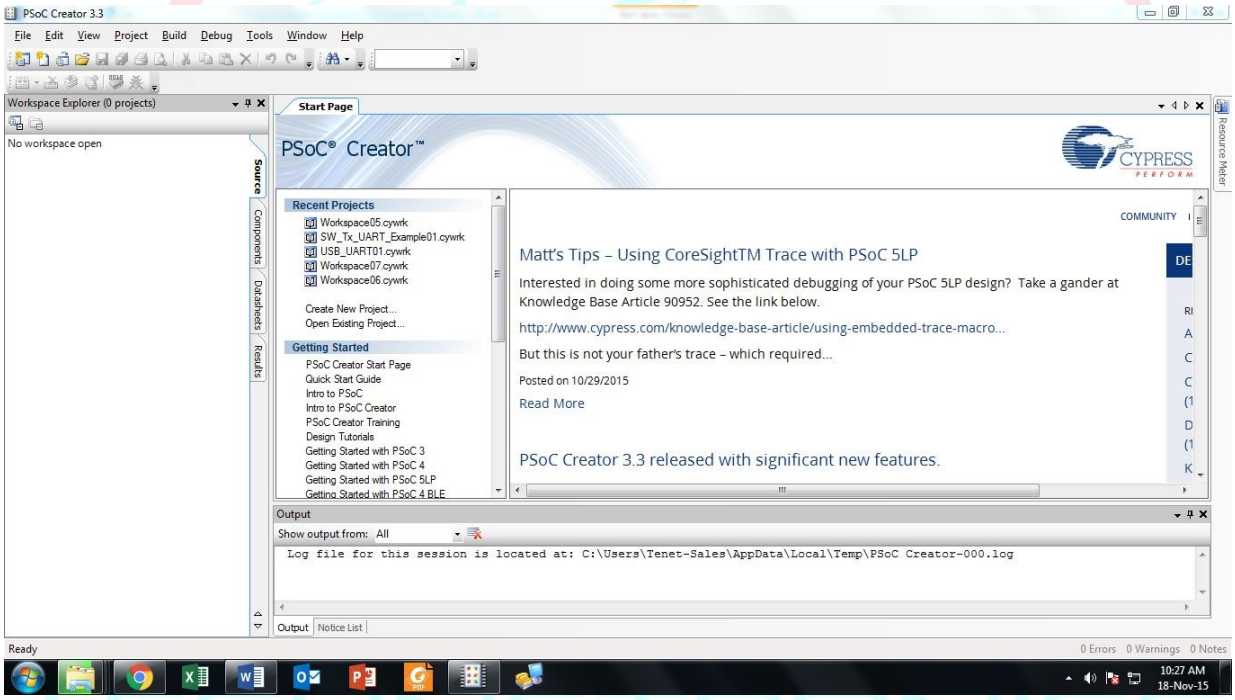


Figure 1

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

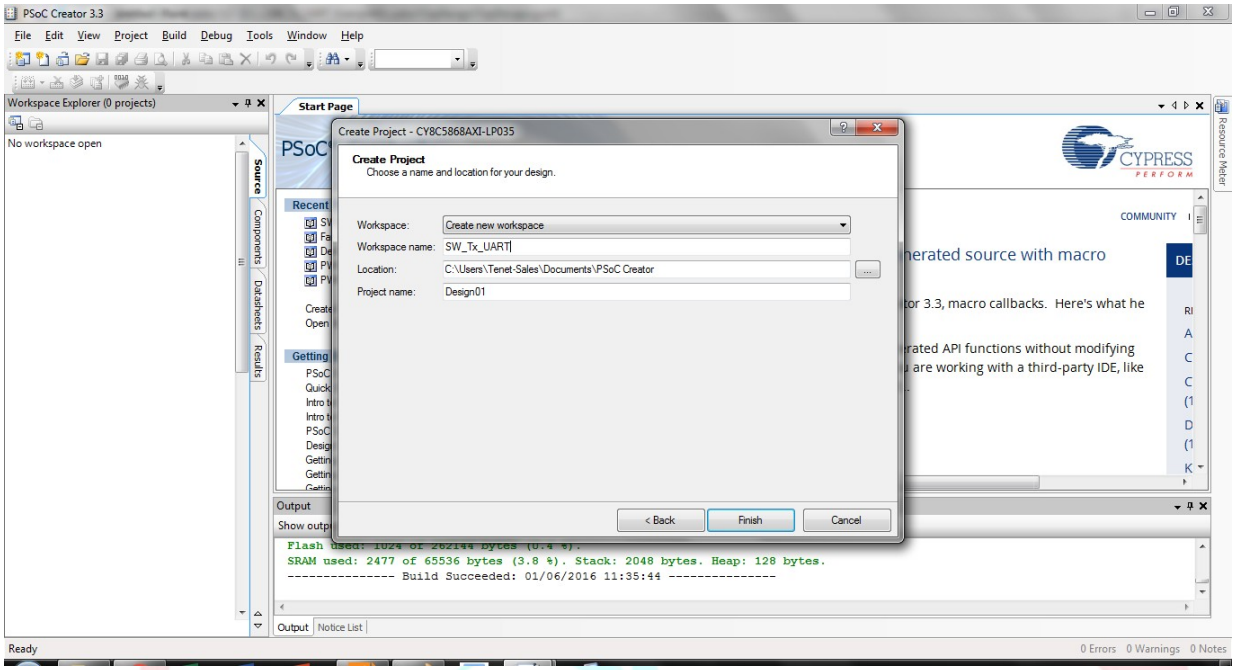


Figure 1

Step 3: Open TopDesign.cysch from workspace explorer.

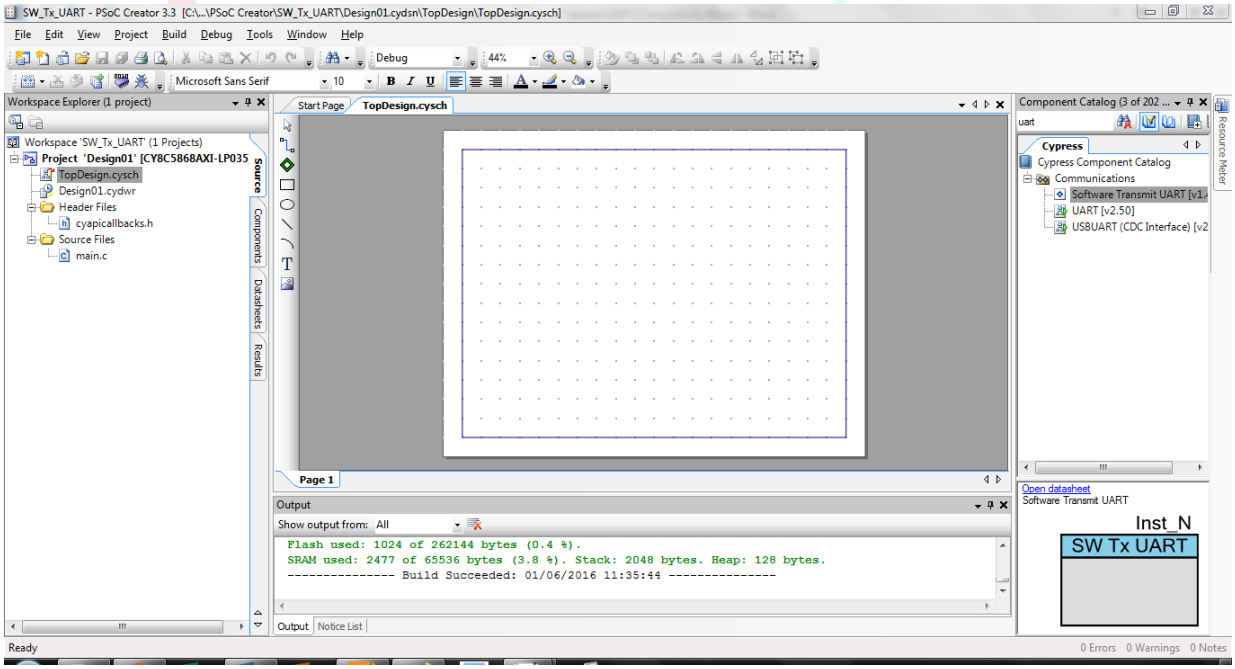


Figure 2

Step 4: Search for Software Transmit UART block from the Component catalog on right side of the window. Drag the Software Transmit UART block onto the workspace. Configure it to 57600 baud rate.

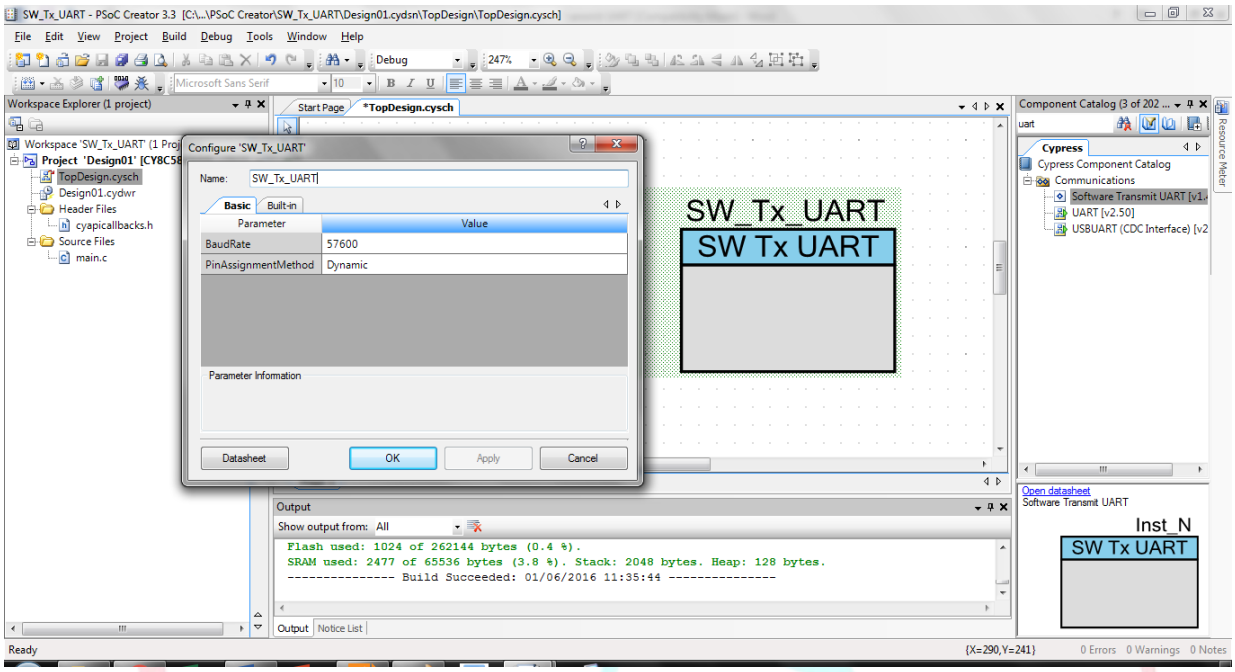


Figure 3

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.

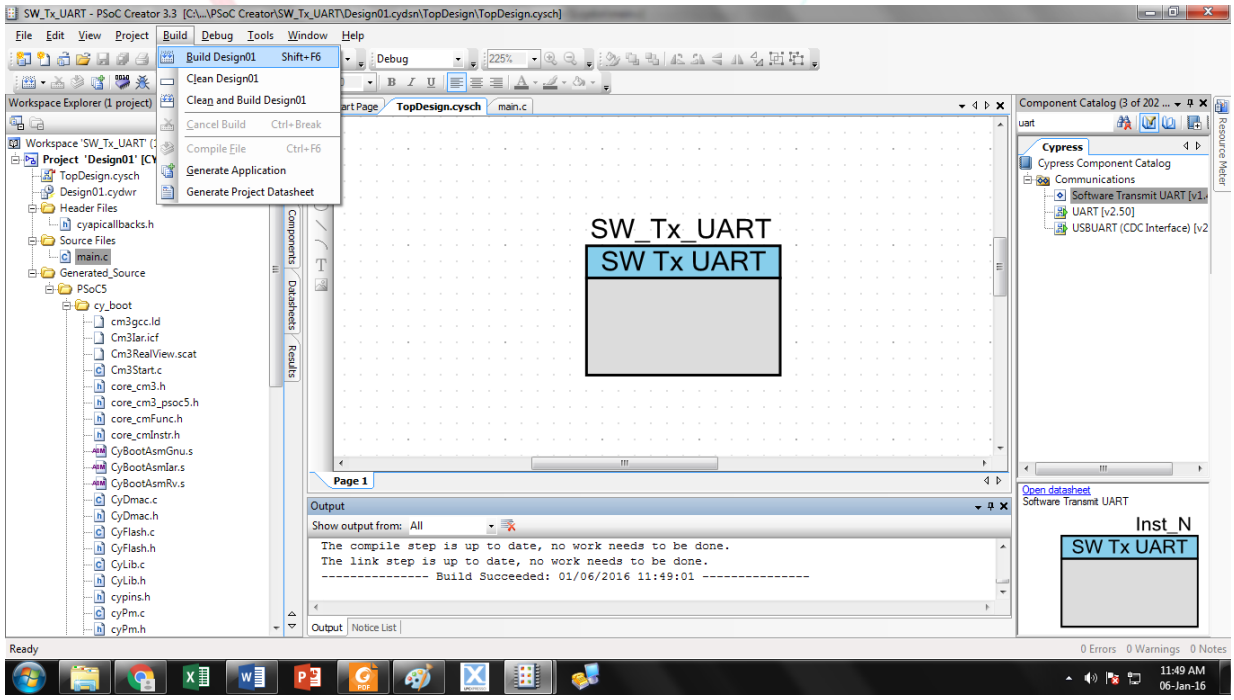


Figure 5

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

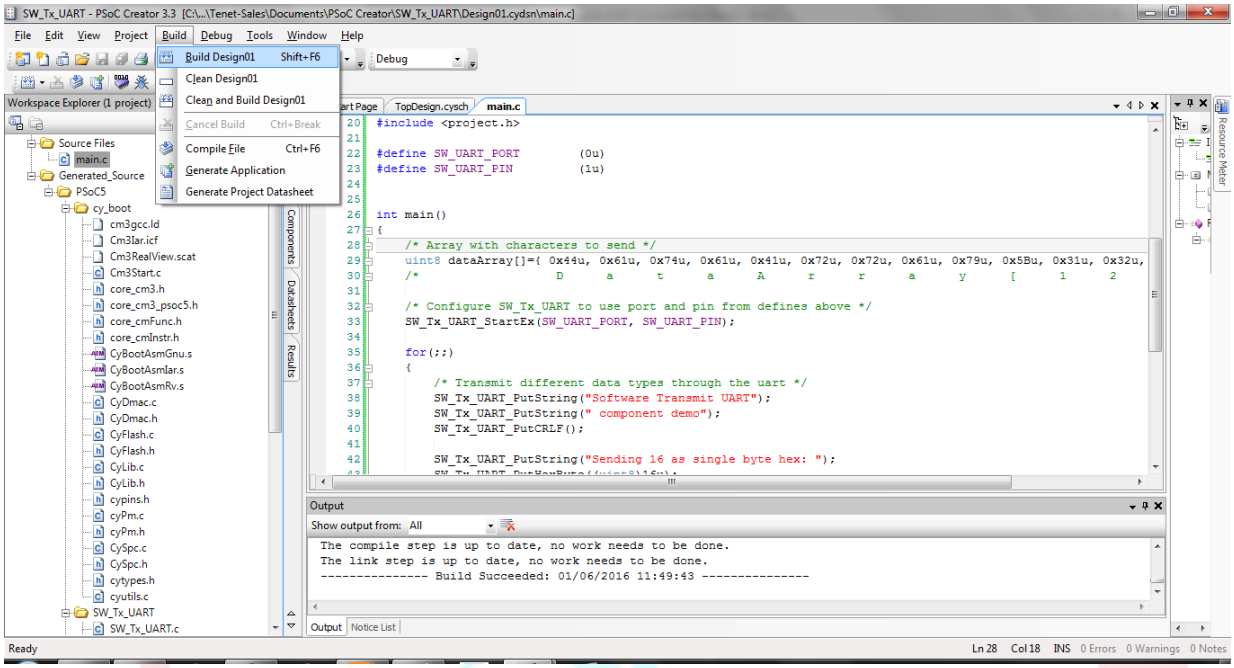


Figure 6

CODE:

```
#include <project.h>

#define SW_UART_PORT          (0u)
#define SW_UART_PIN           (1u)

int main()
{
    /* Array with characters to send (ASCII values)*/
    uint8 dataArray[]={ 0x44u, 0x61u, 0x74u, 0x61u, 0x41u, 0x72u, 0x72u, 0x61u,
0x79u, 0x5Bu, 0x31u, 0x32u, 0x5Du };

    /* Configure SW_Tx_UART to use port and pin from defines above */
    SW_Tx_UART_StartEx(SW_UART_PORT, SW_UART_PIN);

    for(;;)
    {
        /* Transmit different data types through the uart */
        SW_Tx_UART_PutString("Software Transmit UART");
        SW_Tx_UART_PutString(" component demo");
        SW_Tx_UART_PutCRLF();

        SW_Tx_UART_PutString("Sending 16 as single byte hex: ");
        SW_Tx_UART_PutHexByte((uint8)16u);
        SW_Tx_UART_PutCRLF();
```

```
        SW_Tx_UART_PutString("Sending 65534 as two byte hex: ");
        SW_Tx_UART_PutHexInt((uint16)65536u);
        SW_Tx_UART_PutCRLF();

        SW_Tx_UART_PutString("Sending array: ");
        SW_Tx_UART_PutArray(dataArray,13u);
        SW_Tx_UART_PutCRLF();

        CyDelay(3000u);

        SW_Tx_UART_PutCRLF();
    }
}
```

Step 7: Click on Design01.cydwr from Workspace Explorer.

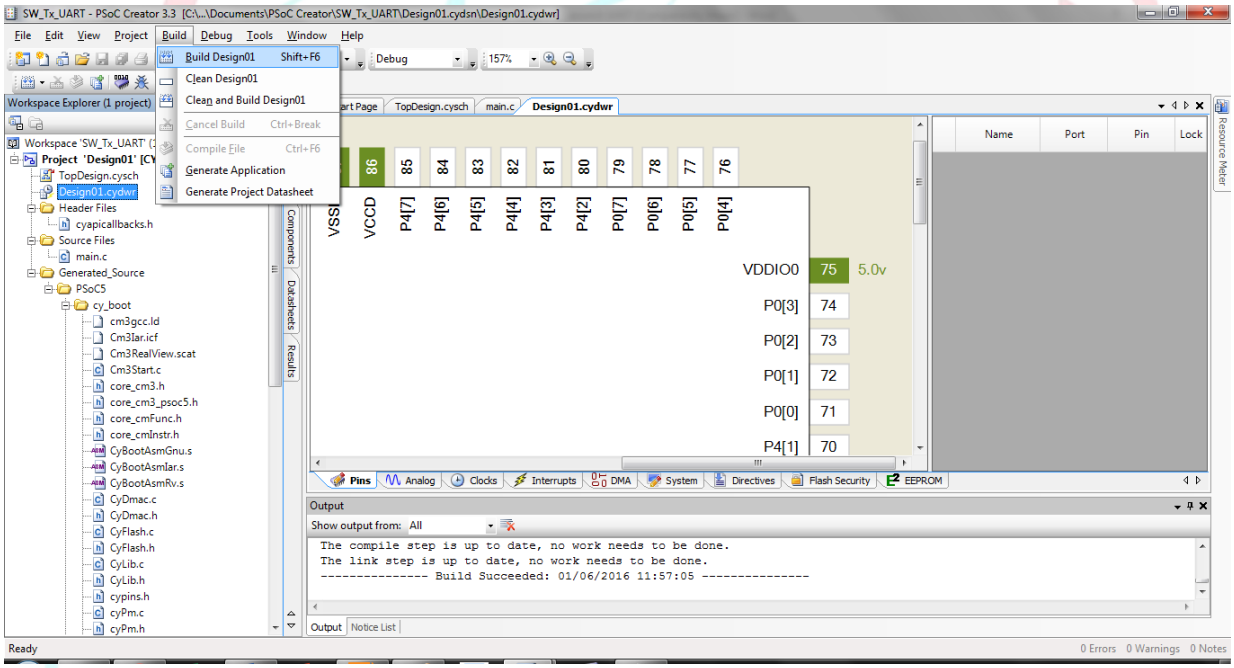


Figure 7

Step 8: If everything goes well connect Micro B cable to debugger port of Freesoc2. To upload the file to Freesoc2 click on Debug --> Program.

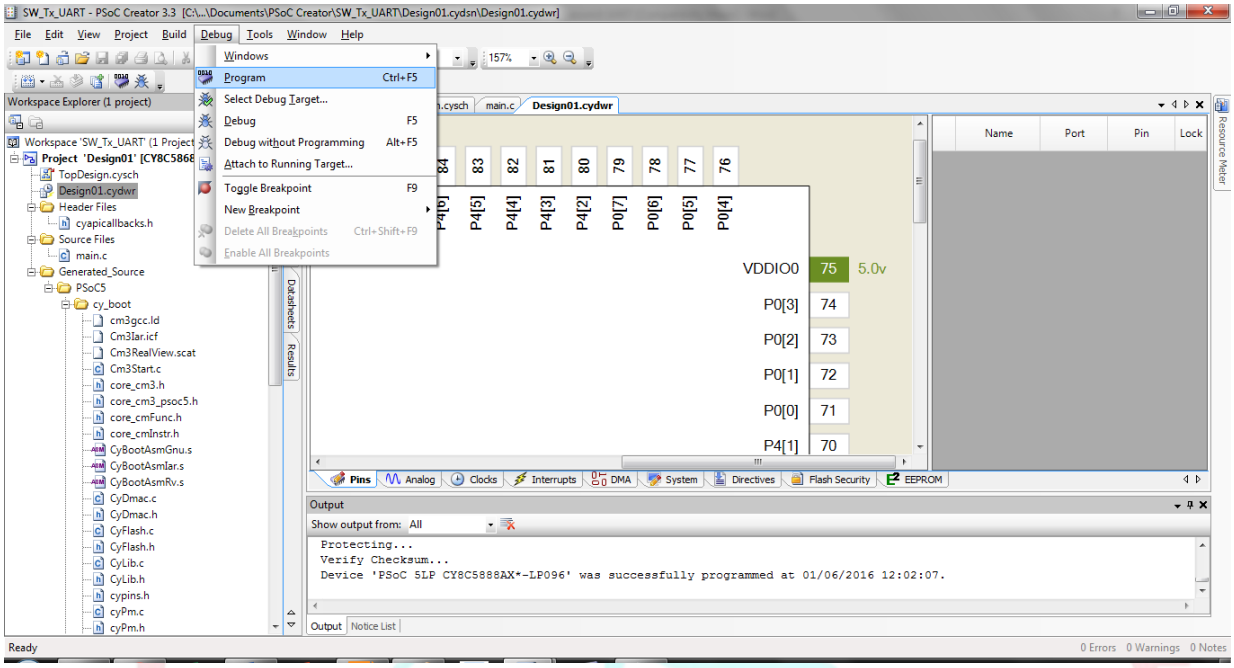


Figure 8

Step 9: Now to display string of characters onto the 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.

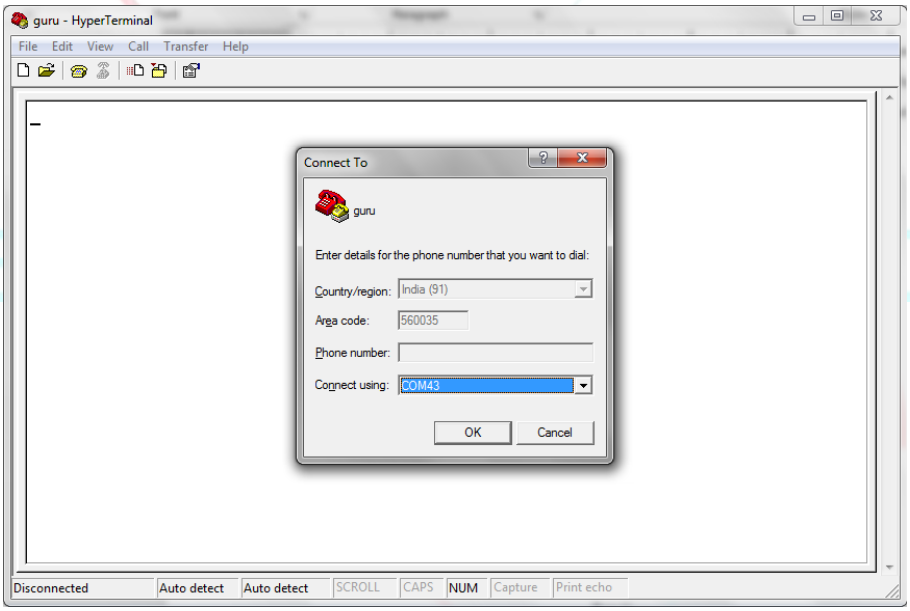


Figure 9

Step 10: Configure as 8-bit and baud rate of 57600 bps.

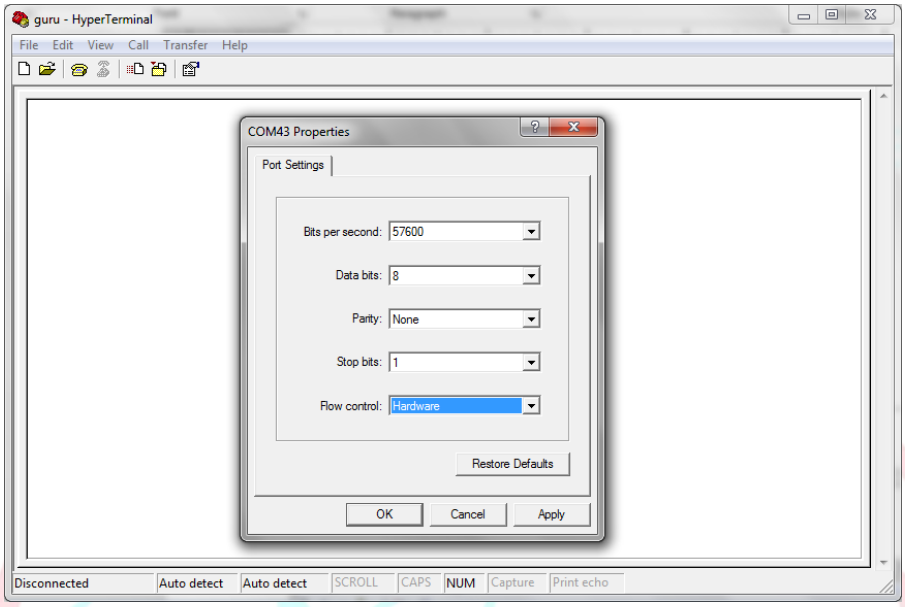


Figure 10

Step 11: This is what we will be able to see.

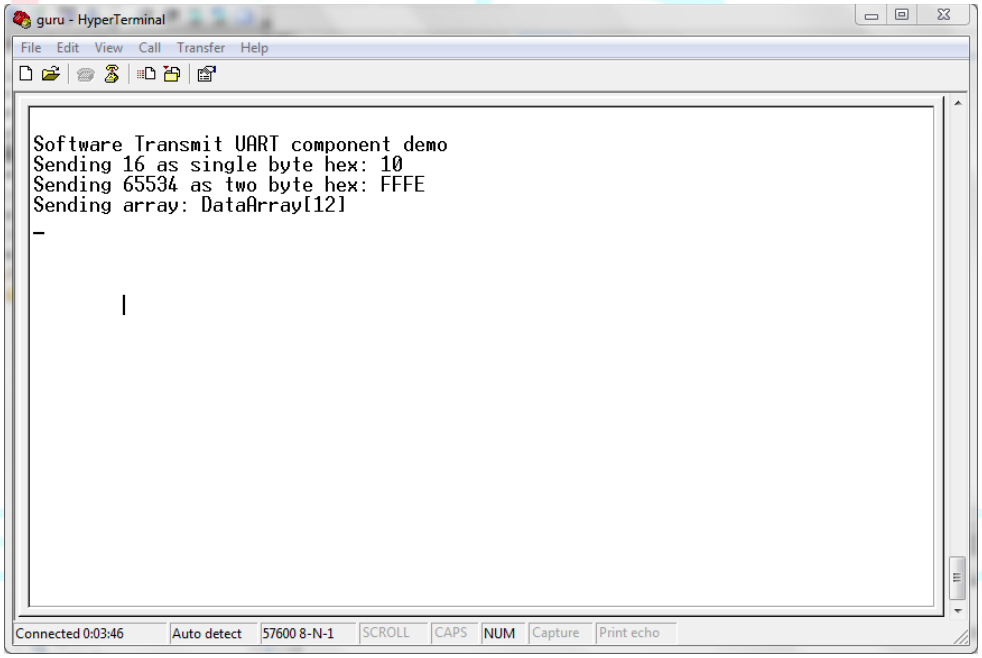


Figure 11

CIRCUIT EXPLAINATION:

The Software Transmit UART module we just added is typically used to achieve a serial communication. 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.

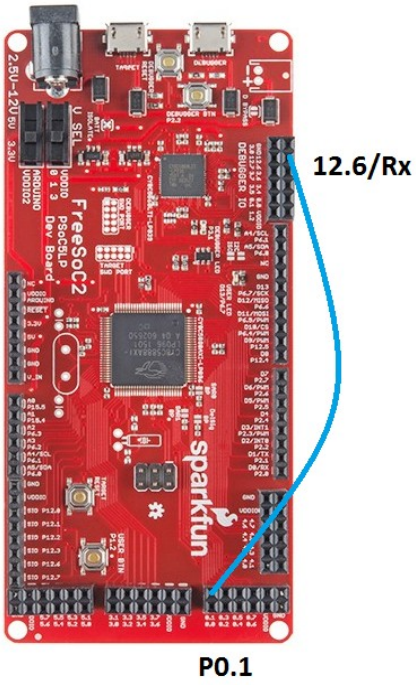


Figure 12

In this project we have configured the Software Transmit via P0.1. RX pin on the debugger/programmer which is 12.6 (RX), receives the transmitted string of characters and displays on the Hyper Terminal.

For product link:

1. **http://www.tenettech.com/product/7241/freesoc2-development-board-psoc5lp**

For more information please visit: www.tenettech.com

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