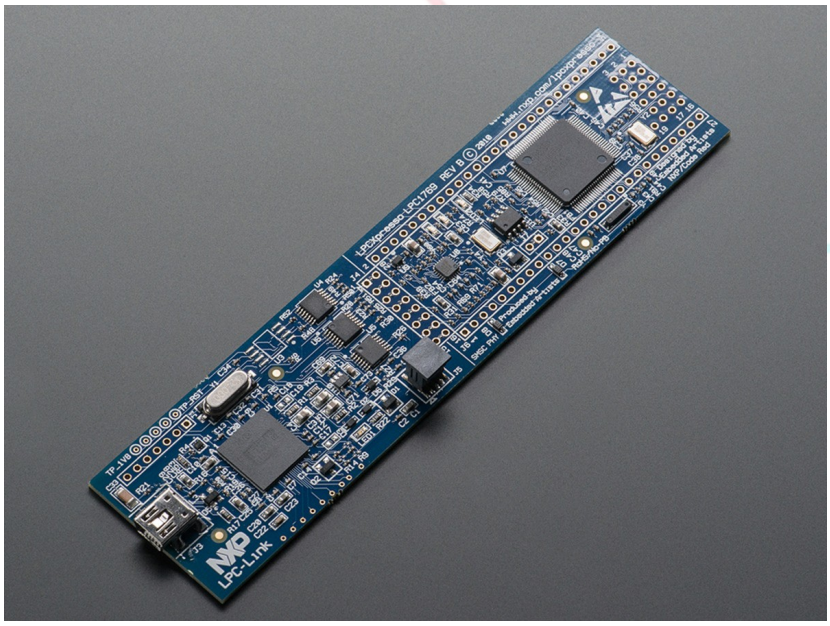
TENET TECHNETRONICS

2016

# UART(Serial Communication) with NXP LPC1769 using LPCXpresso



*Author: Gurudatta Palankar*

*Reviewers:*

*Version: 1.0*

# 9/3, 2nd floor, SreeLaksmi Complex, opp, to Vivekananda Park, Girinagar, Bangalore - 560085,
Email: info@tenettech.com, Phone: 080 - 26722726

## Introduction:

**LPCXpresso™ is a new, low-cost development platform available from NXP supporting NXP's ARM-based microcontrollers. The platform is comprised of a simplified Eclipse-based IDE and low-cost target boards which include an attached JTAG debugger. LPCXpresso™ is an end-to-end solution enabling engineers to develop their applications from initial evaluation to final production.**

## Step 1: Open LPCXpresso IDE

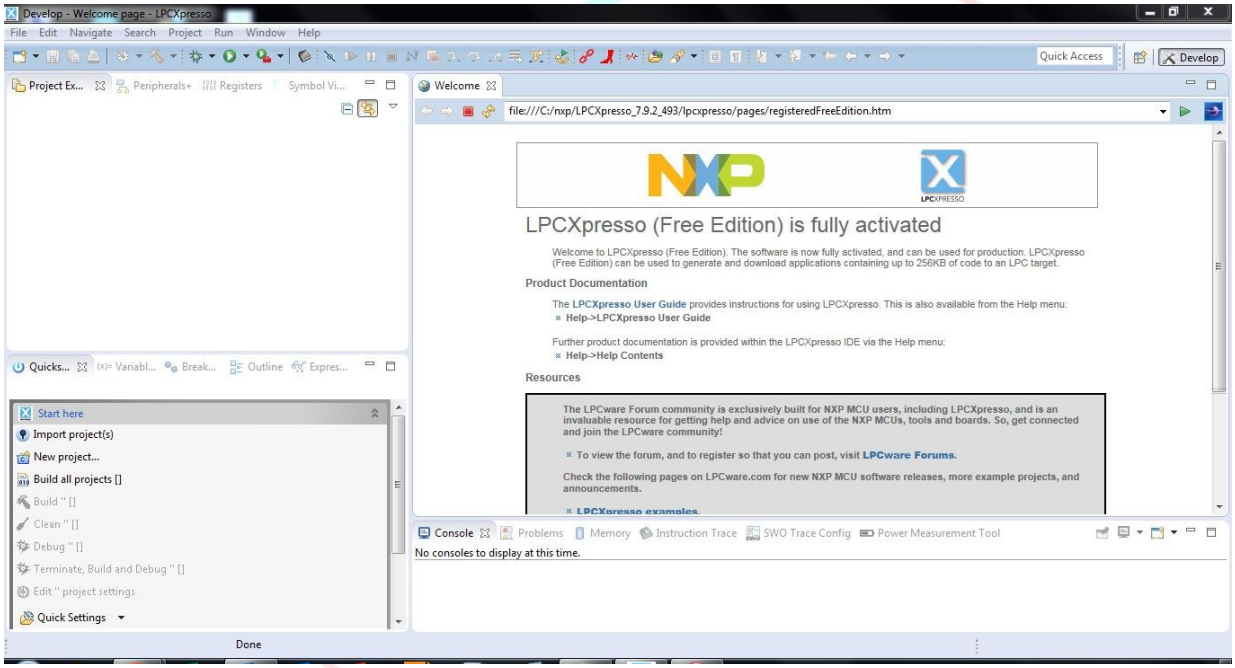



Figure 1

**Step 2: Before writing a code, we have to Import some Library Files to the Workspace. Click on Import projects on Quickstart Panel on the bottom left of the window.**

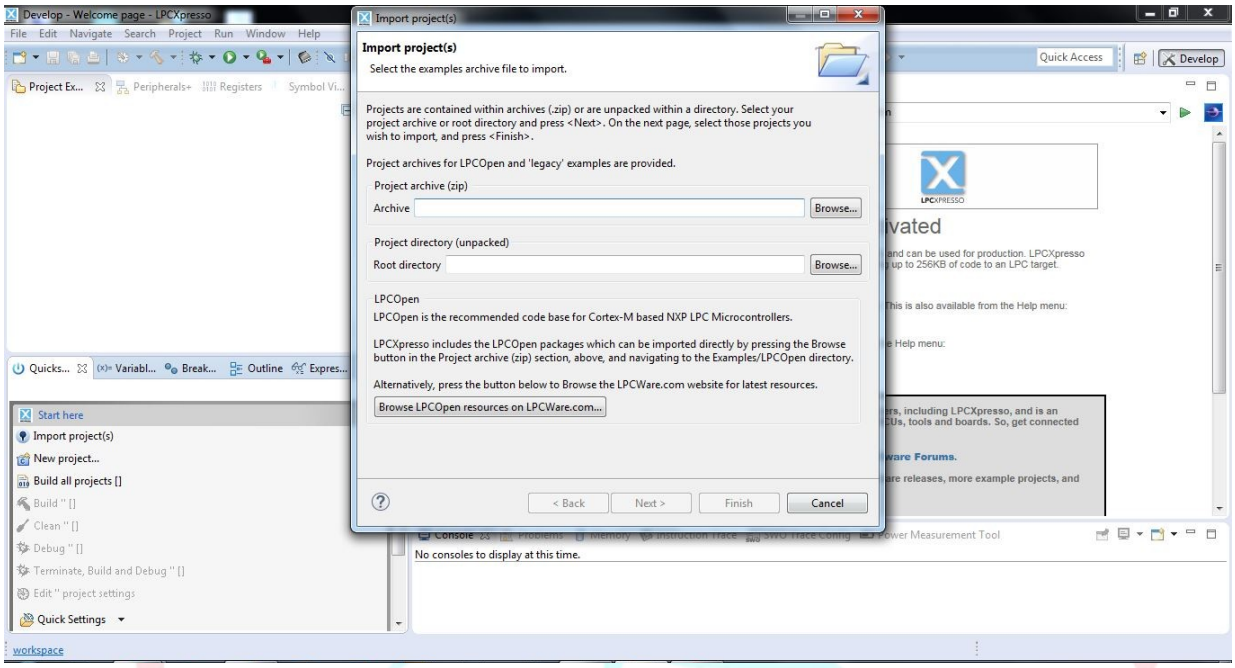



Figure 2

**Step 3: Browse file, open the LPC1000 folder.**

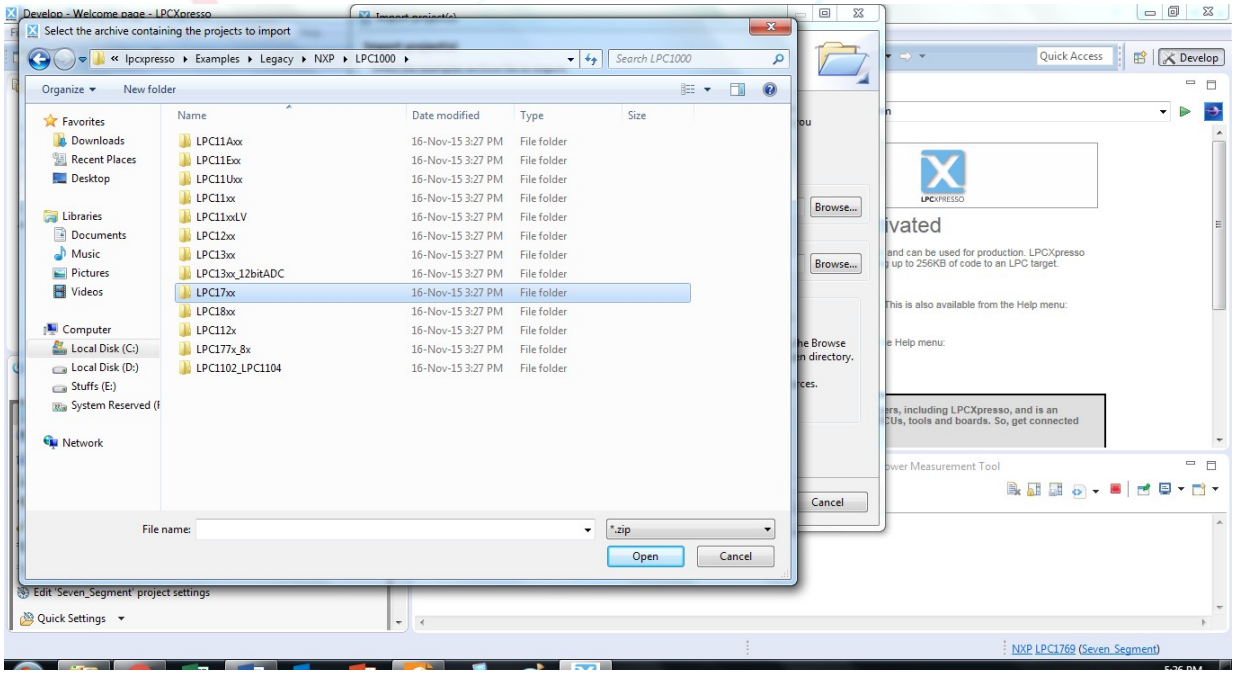



Figure 3

**Step 4: Select the appropriate archive file. Let us select LPCXpresso176x_cmsis2. We can select CMSIS CORE library that include LPC17xx.h header file.**

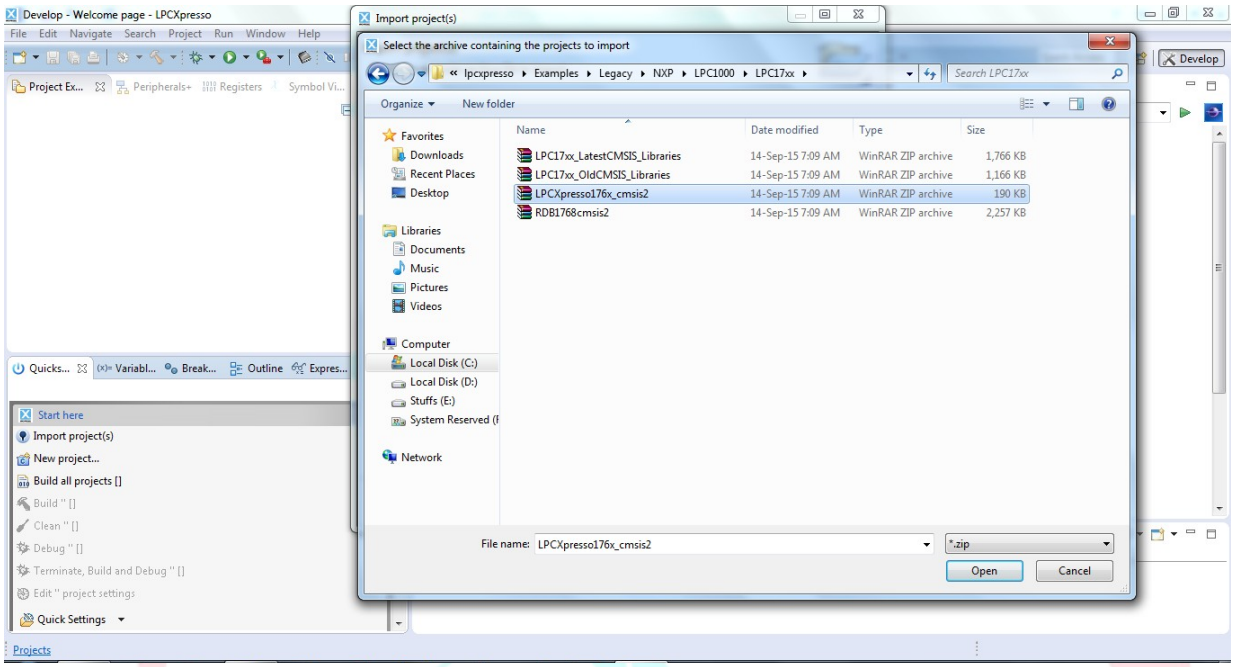



Figure 4

**Step 5: After selecting you will be able to see the following libraries files. Let us select specific one.**

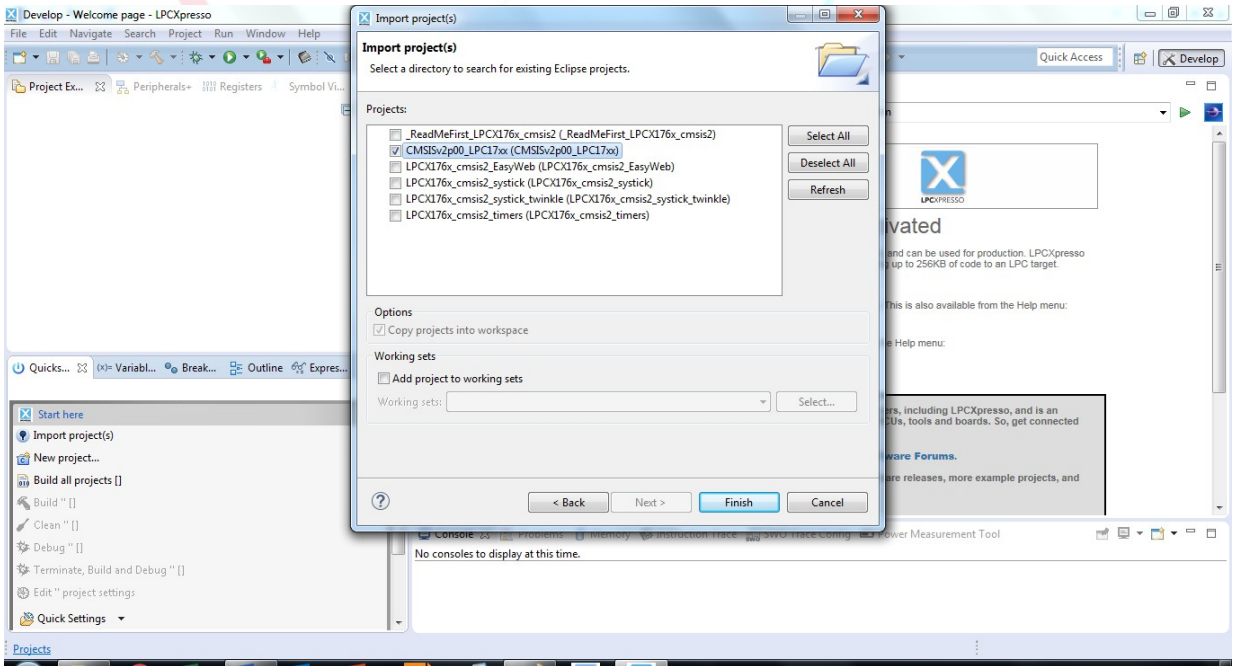



Figure 5

**Step 6: Now we will be able to see those libraries in the workspace.**

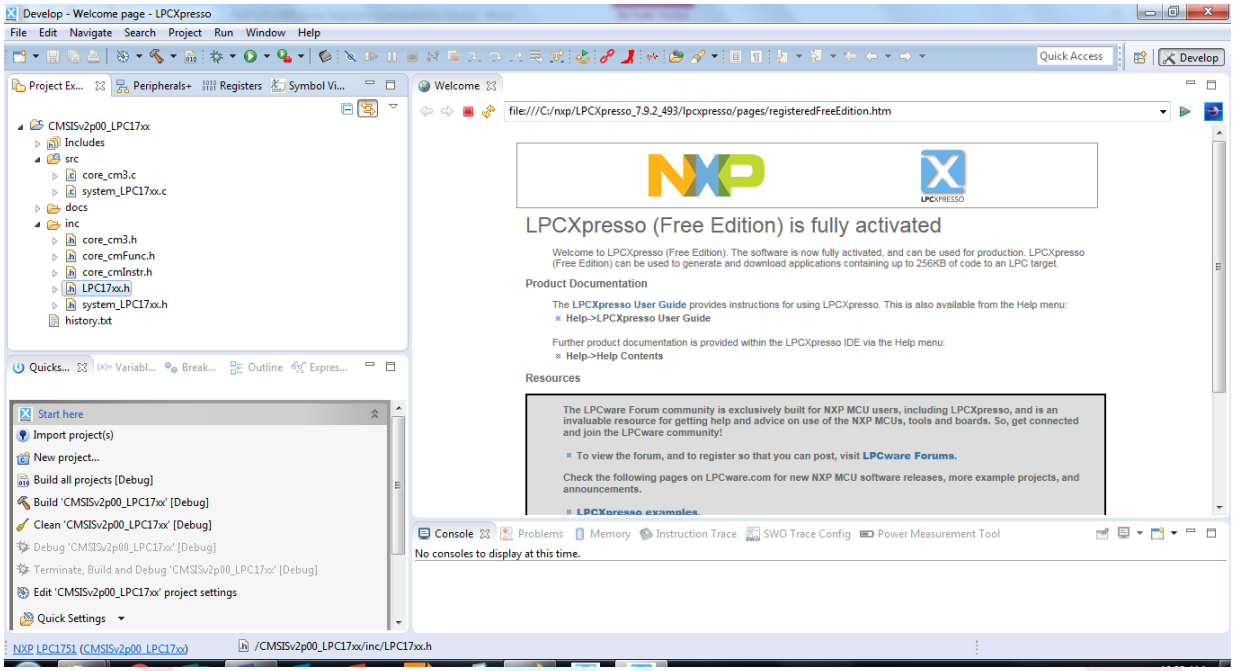



Figure 6

**Step 7: Now we can start creating our new project. Goto File >> New >> Project. Select LPCXpresso C project.**

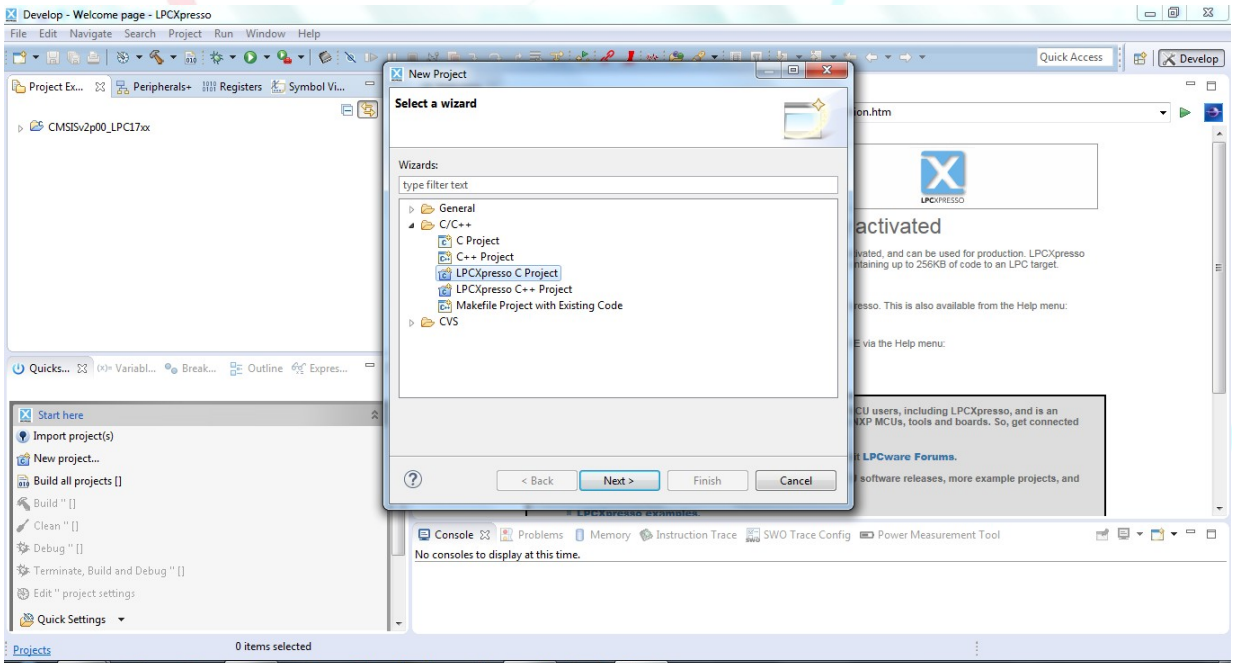



Figure 7

**Step 8: Select LPC1769, C Project and give name to your project. Select target MCU as LPC1769.**

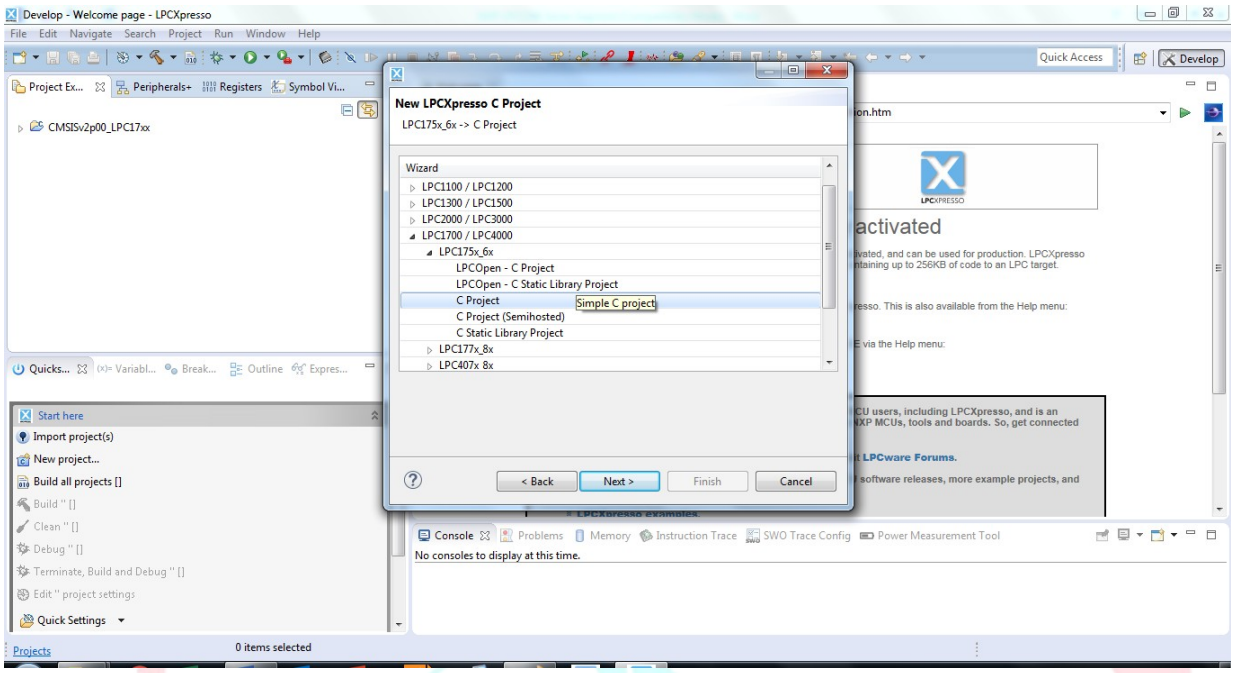



Figure 5

**Step 9: Now select CMSIS Core library. Click on Next and keep all the other configurations as default and Finish.**

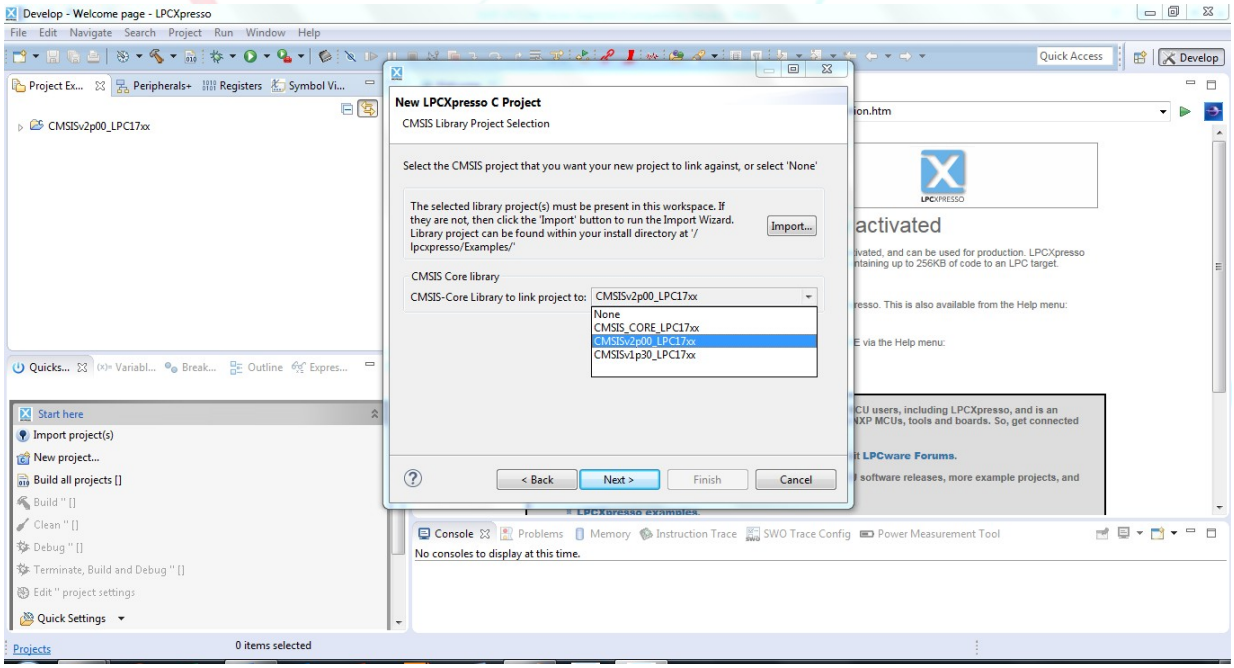



Figure 9

**Step 10: Now we can see our project onto the workspace. Now by double clicking on LCD_display.c file, we can start writing code.**

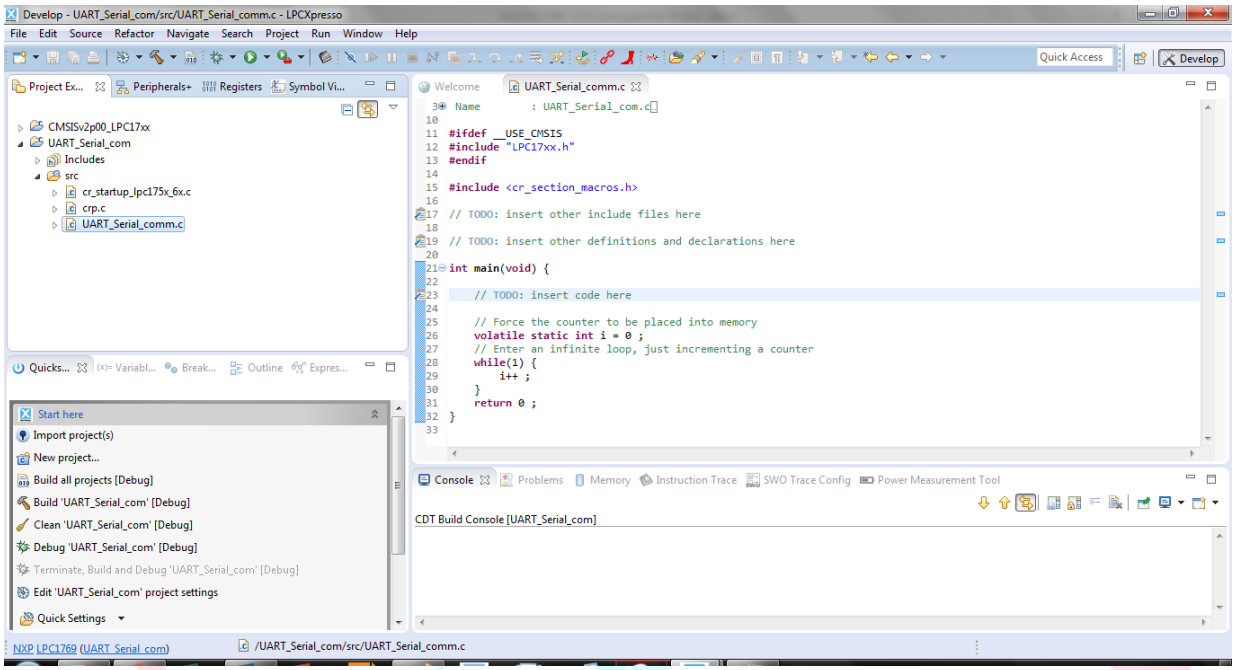



Figure 10

**Step 11: Write a code as shown below.**

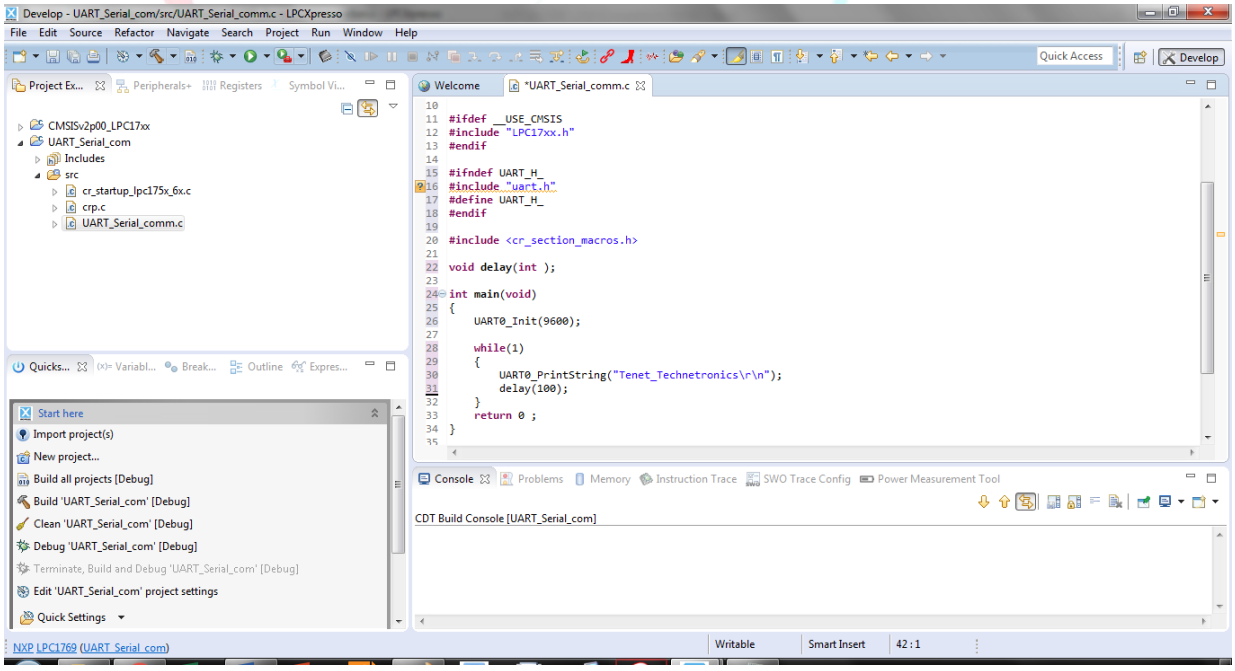



Figure 6

**CODE:**

```
#ifdef __USE_CMSIS
#include "LPC17xx.h"
#endif

#include "uart.h"

void delay(int );

int main(void)
{
    UART0_Init(9600);

    while(1)
    {
       UART0_PrintString("Tenet_Technetronics\r\n");
       delay(100);
    }
    return 0 ;
}

void delay(int a)
{
      int i,j;
      for(i=0; i<500000; i++)
      for(j=0; j<a; j++);
}
```

**NOTE:** **The above code will not work until and unless we add Header File (uart.h) and Source File (uart.c) to the project.**

## Step 12: To create or add library files, right click on src file of your project file, then New >> Source File.

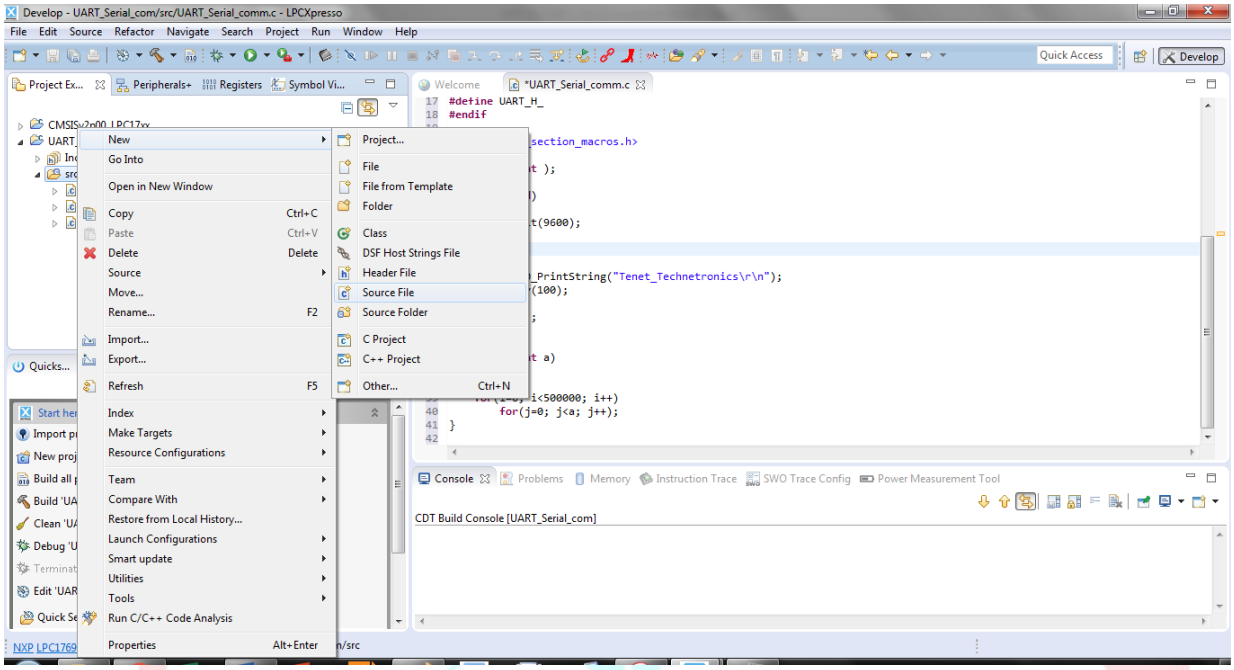



Figure 7

## Step 13: Save the Source File name with .c file extension.

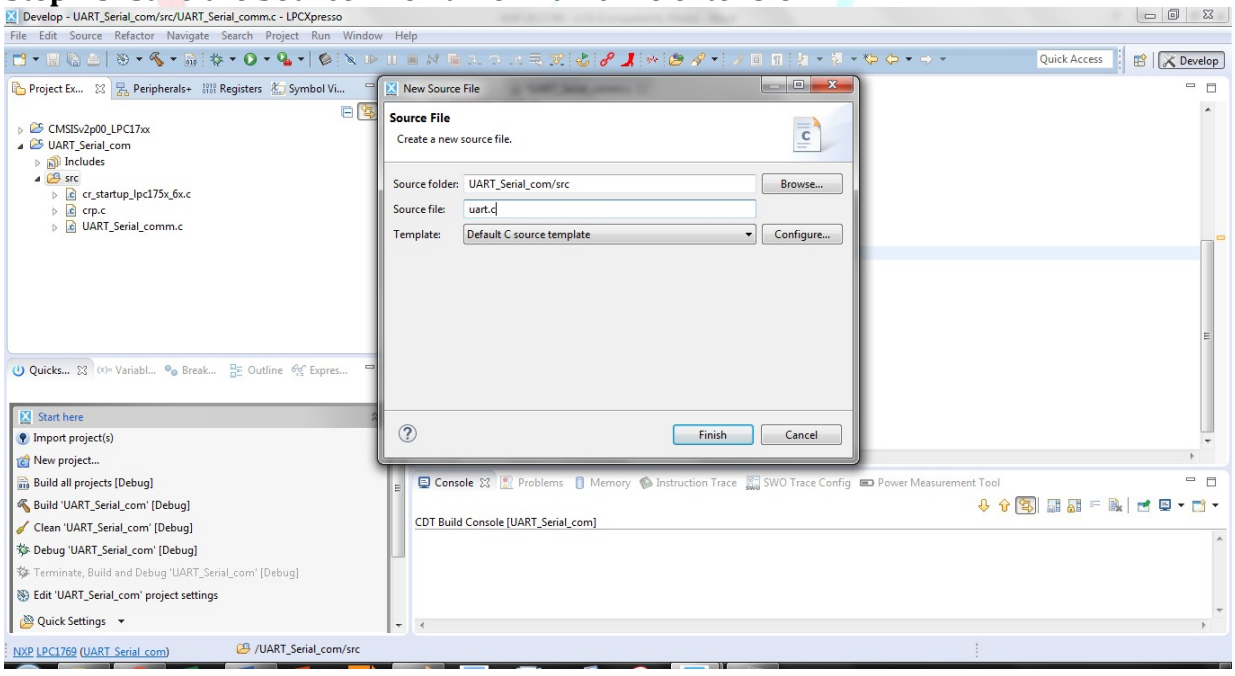



Figure 13

## Step 14: Write or paste the Source File (Library).

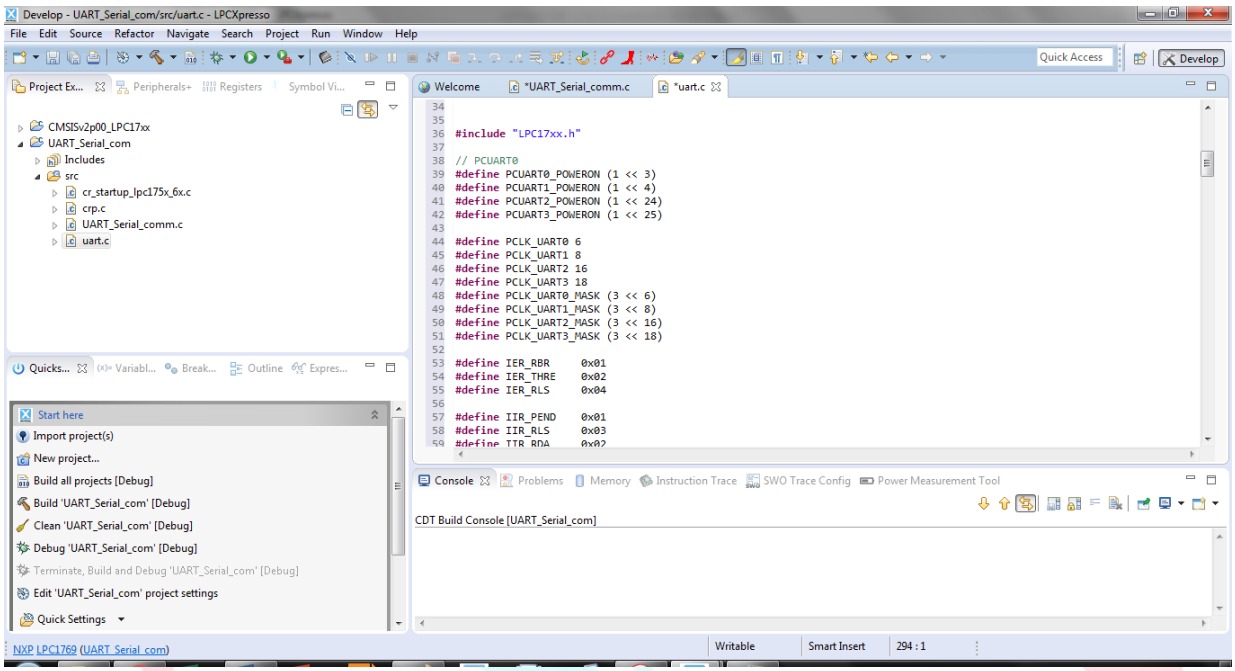



Figure 14

## Step 15: Similarly add the Header File.

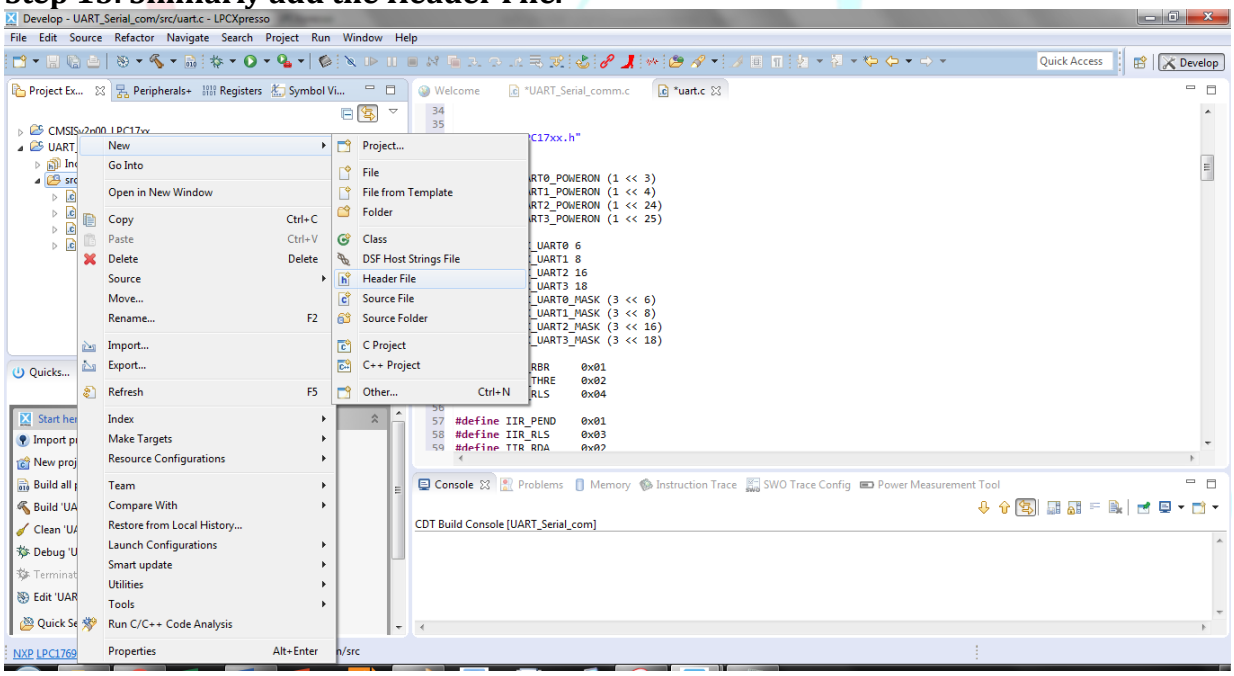



Figure 15

## Step 16: Save the Header File name with .h file extension.

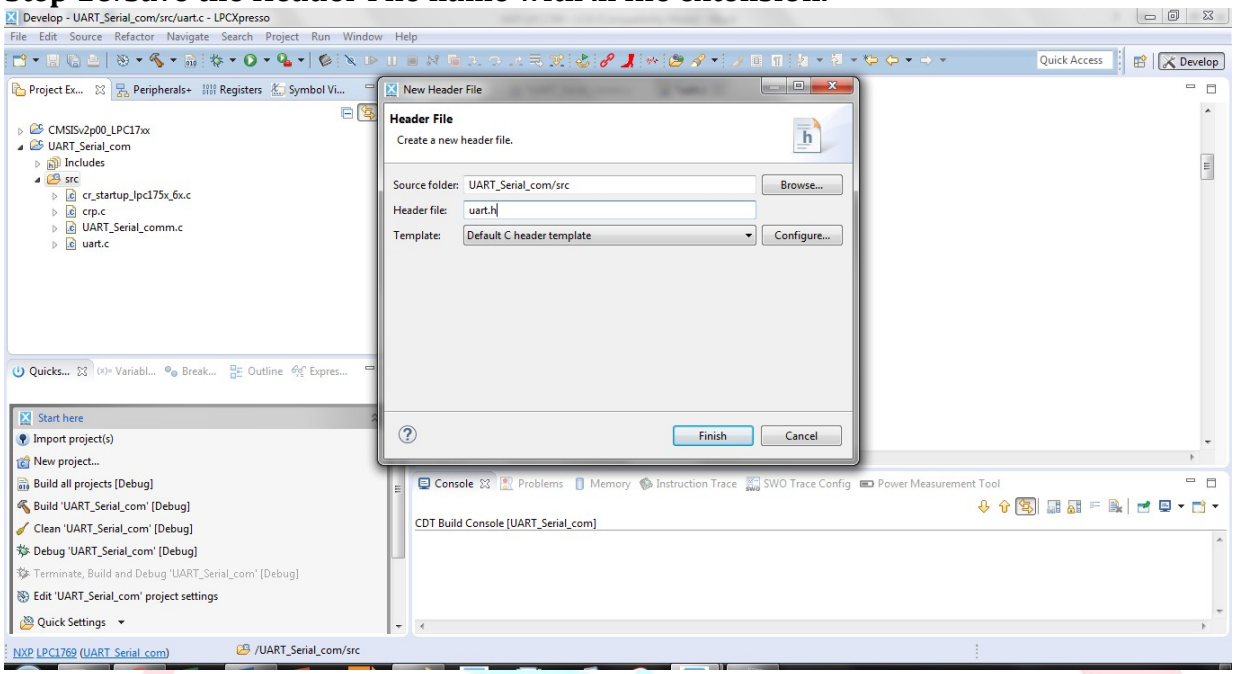



Figure 16

## Step 17: Write or paste the Header File (Library).

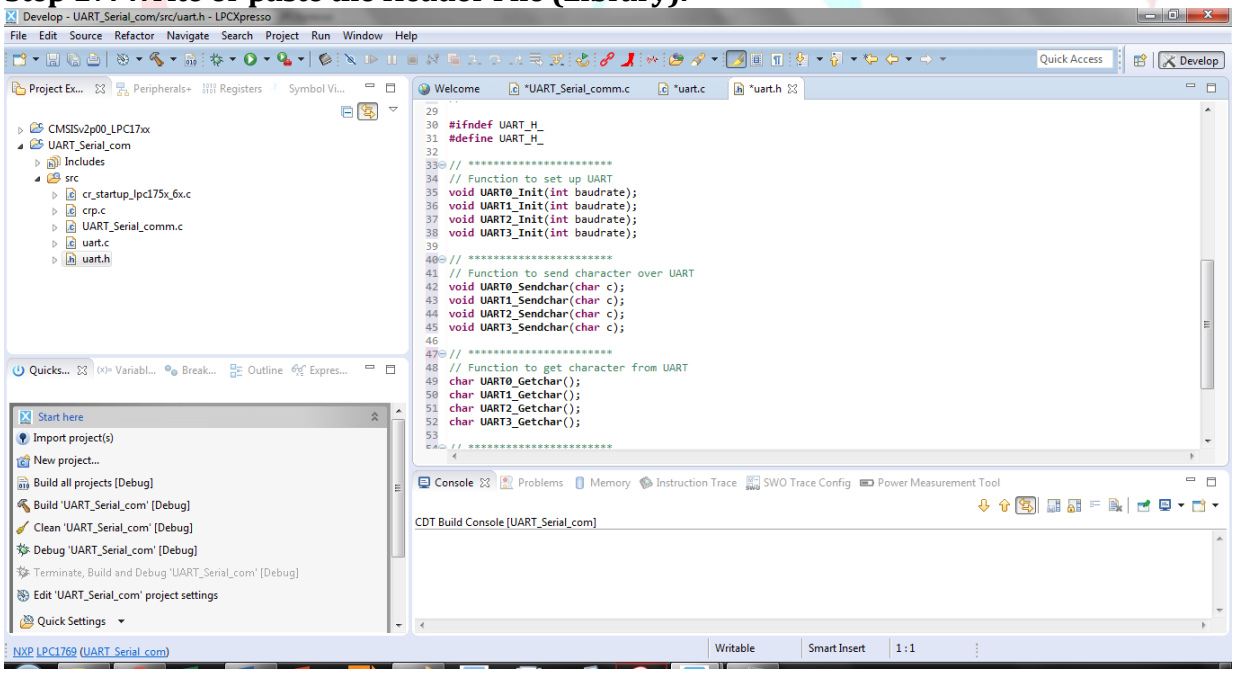



Figure 17

**Step 18: After writing code and adding libraries, Build the project by clicking on Build UART_Serial_comm.c on the Quickstart Panel on the bottom left of the window.**

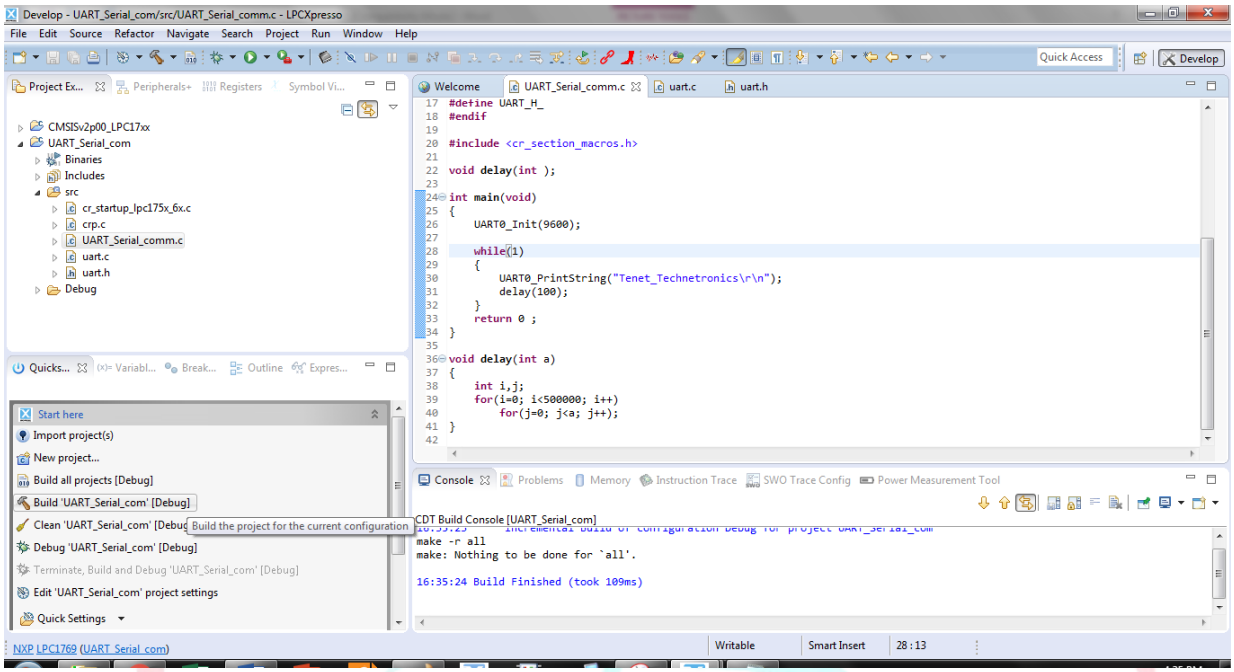



Figure 18

**Step 19: Now, if all goes well connect the Micro B cable to LPC1769 and connect it to your computer. To upload the project file, click on the Program flash.**

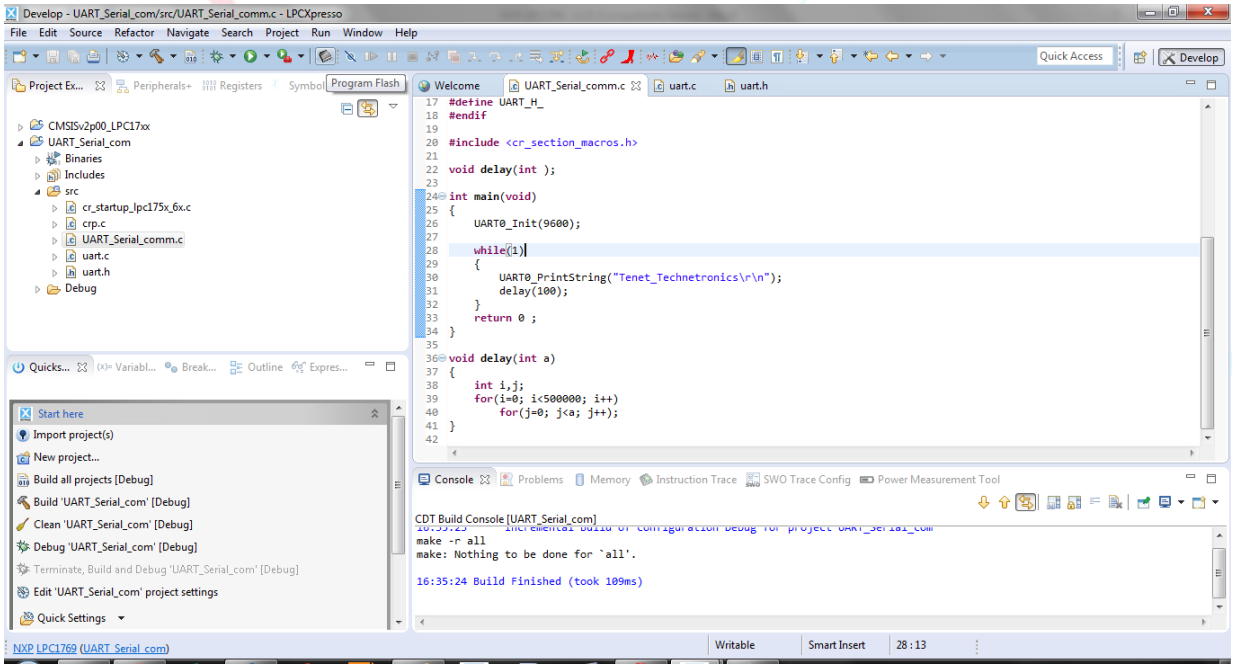



Figure 19

**Step 20: Now select the Project file UART_Serial_comm.axf. We can find it in our project folder.**

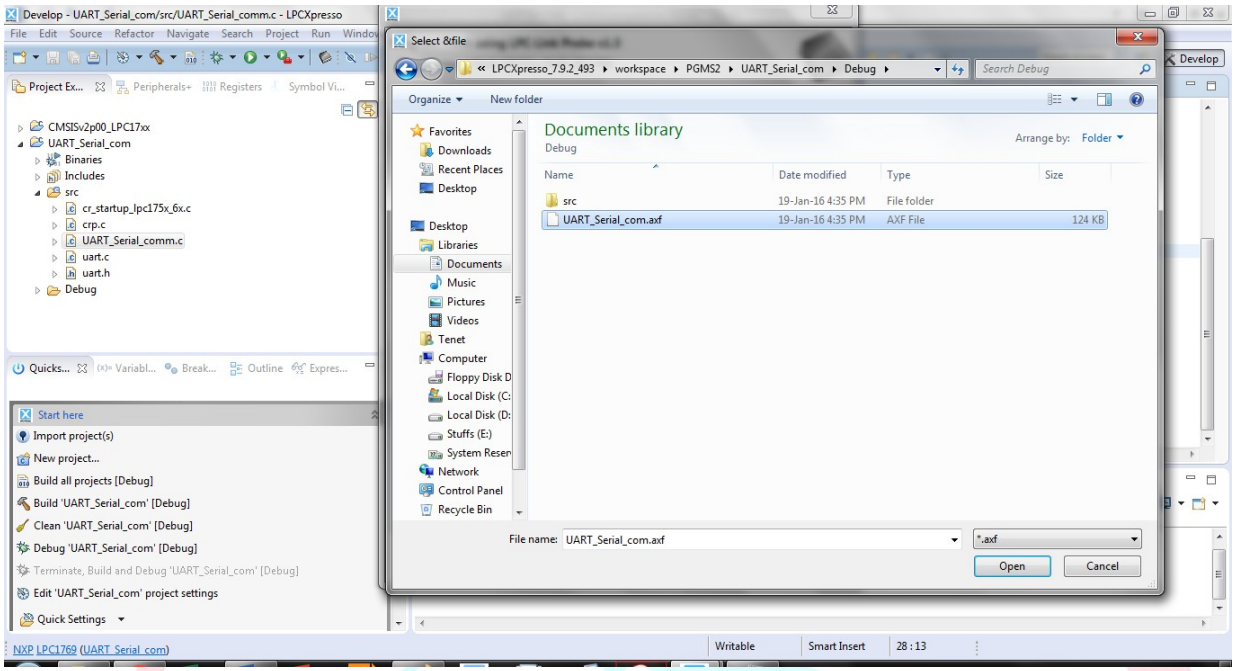



Figure 20

**Step 21: Now this window shows we have finally dumped our project onto LPC1769.**

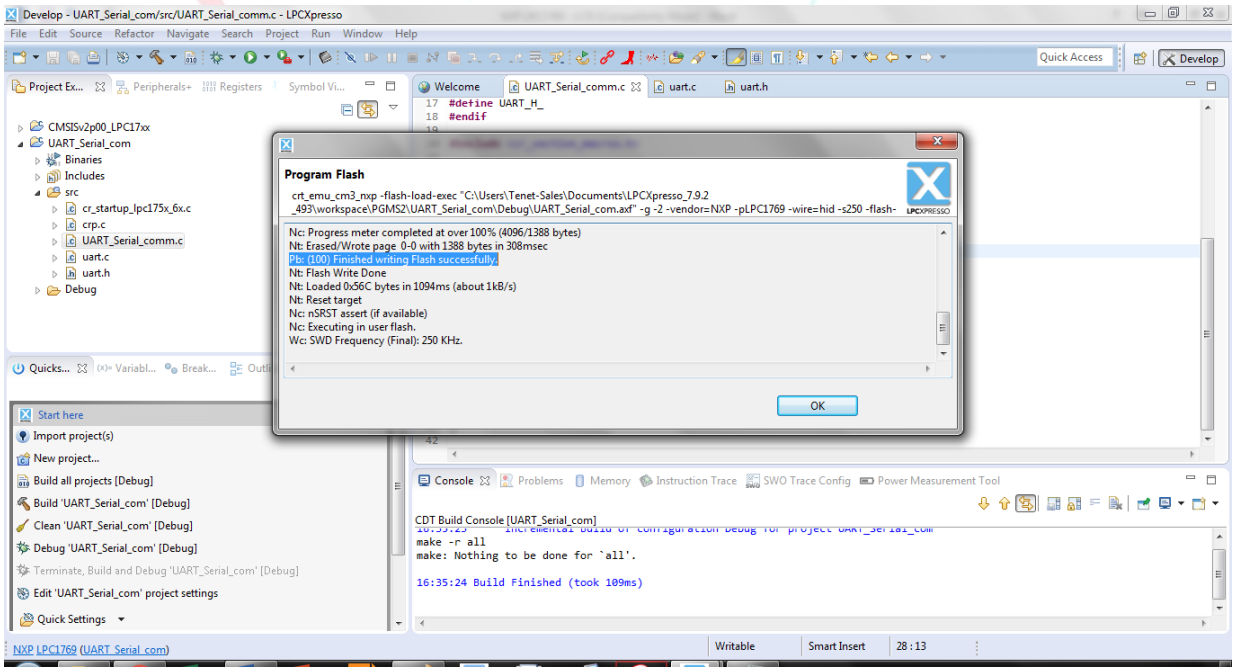



Figure 21

*Hardware Requirements:*

- **LPC1769 Board**
- **Jumper wires**

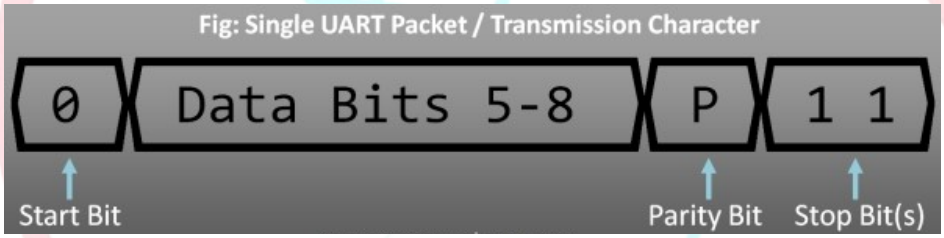
*UART basics:*

**Whenever we want to communicate between PC and MCU or between two MCUs, the simplest way to achieve that is using UART. UART stands for Universal Asynchronous Receiver/Transmitter.**

**UART communication basically uses 2 pins for Data transfer and these are:**

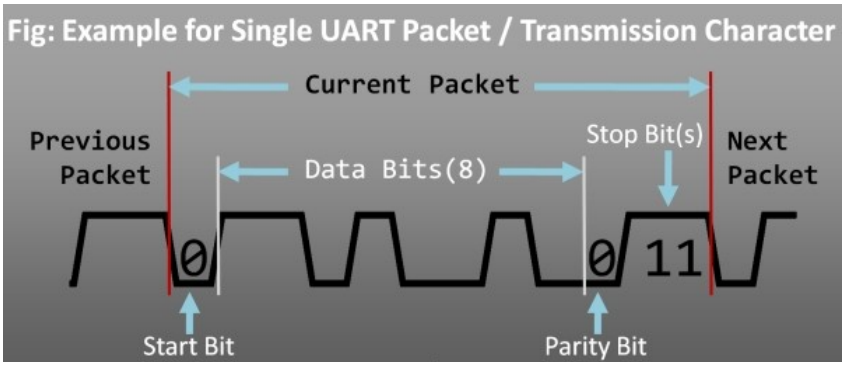
1. **TxD (or Tx) – which is the Transmit Data Pin used to send data**
2. **RxD (or Rx) – which is the Receive Data Pin used to get data**

**UART sends & receives data in form of chunks or packets. These chunks or packets are also referred to as 'transmission characters'. The structure of a data packets is as shown below:**

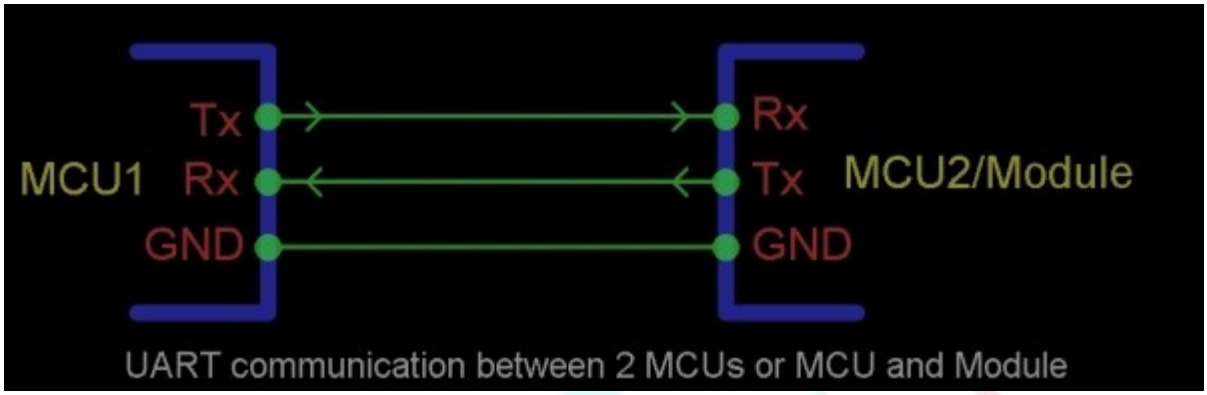



**UART data packet begins with a '0'. This bit is also called as "Start Bit" which signifies incoming data. Next comes the actual data which can be 5 to 8 bits in length. After the data an optional parity bit can be used for error checking. Lastly comes the "Stop Bit(s)" which is a '1' and this signifies end of current data packet. Note that either 1 or 2 stop bits can be used and the parity bit can be : Even , Odd , forced to 1 i.e. Mark parity , forced to 0 i.e. Space parity or None. (In UART/RS232 context a MARK means 1 and SPACE means 0, hence marking state means a stream (series) of 1s and Spacing state means a stream of 0s)**

**Here is an example of a packet having 8 data bits, odd parity bit and 2 stop bits:**



## 1. UART communication between 2 MCUs or MCU and Module




**OUTPUT:**

**UART communication between two micro-controllers, LPC1769 and FreeSoC2 (PSoC5LP).**

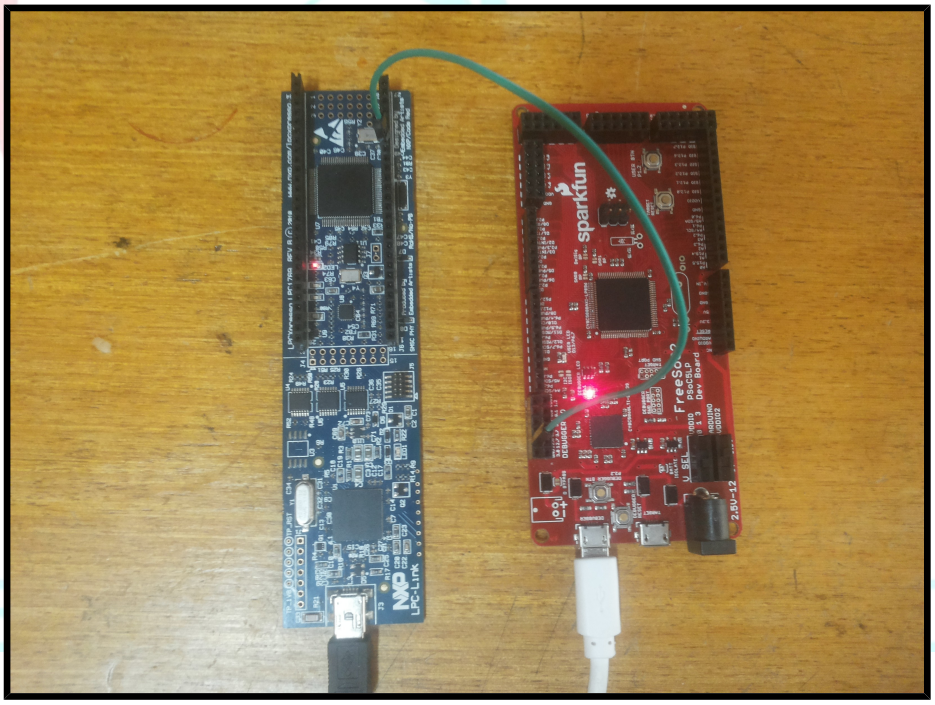
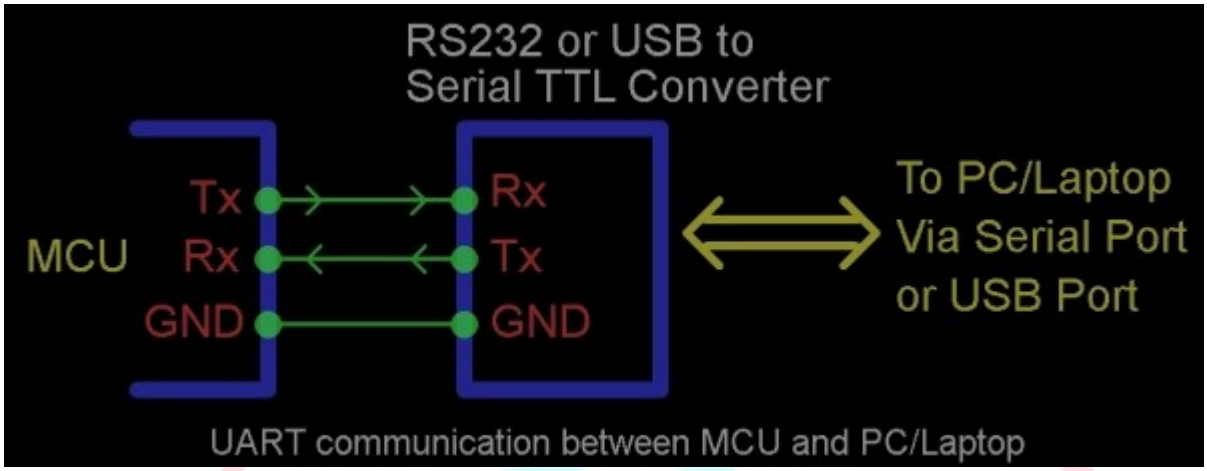


Figure 18

**NOTE: To have Serial Communication between two microcontrollers, we have to program the second microcontroller to receive the transmitted signal.**

**2. UART communication between MCUs or Modules and PC/Laptop**




**USB to Serial TTL converter (Prolific cable)**

**GREEN - TX**

**WHITE - RX**

**RED - No Connection**

**BLACK - GND**

**The output can be viewed in a Serial monitors like HyperTerminal, Putty, XCTU and so on...**

**For product link:**

1. http://www.tenettech.com/product/1548/lpc1769-lpcxpresso-board
2. 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**