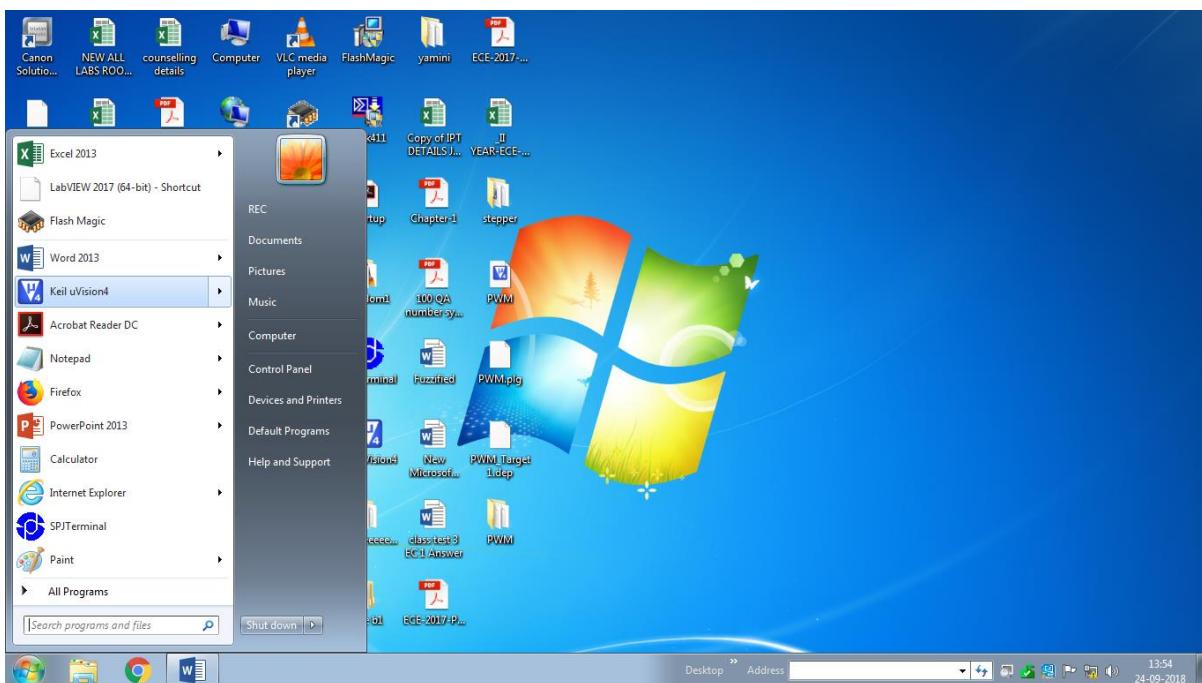
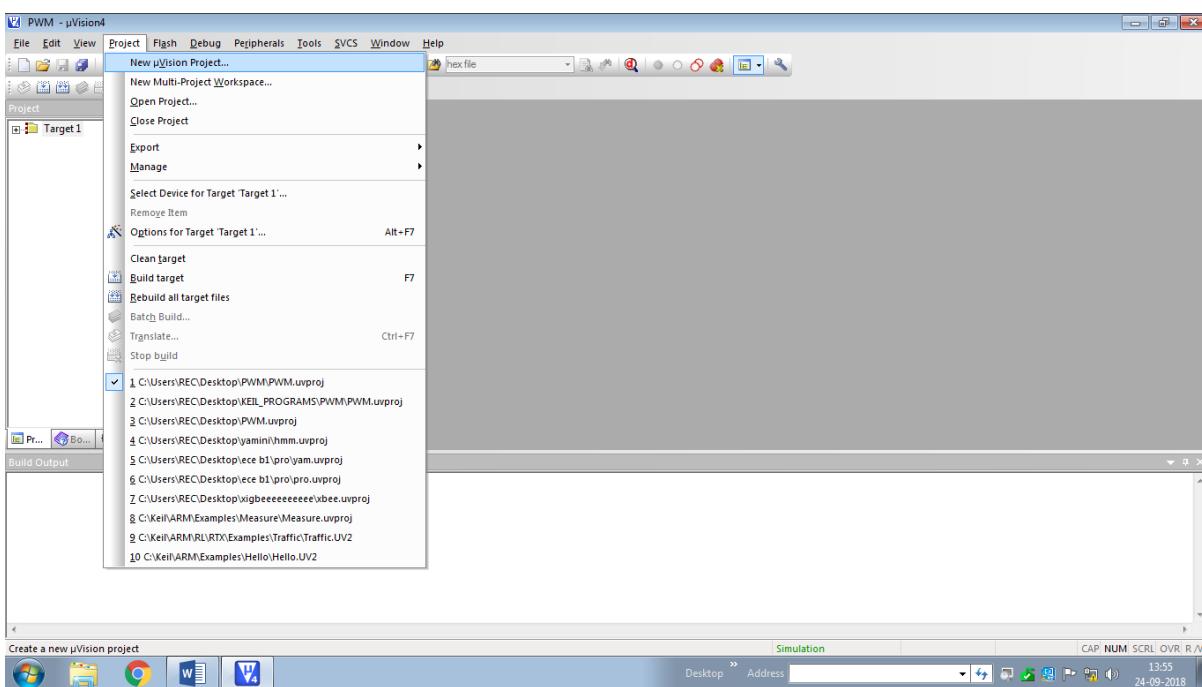


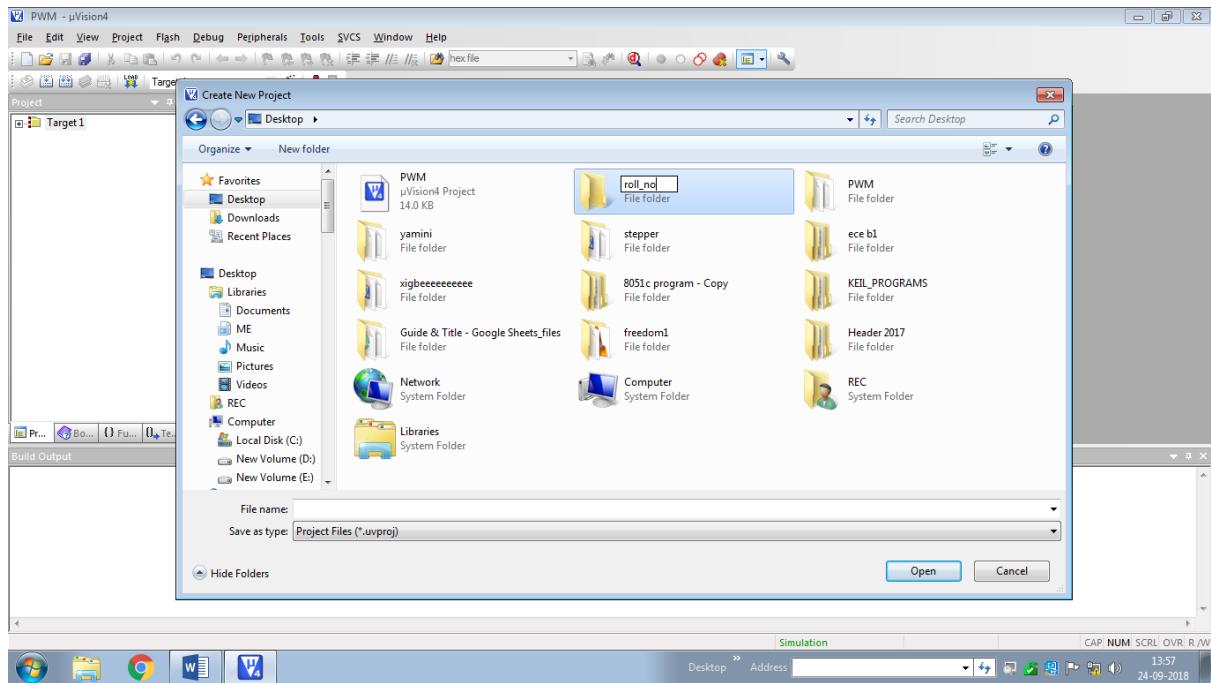
## 1. Select the Keil uVision4 from Start menu.



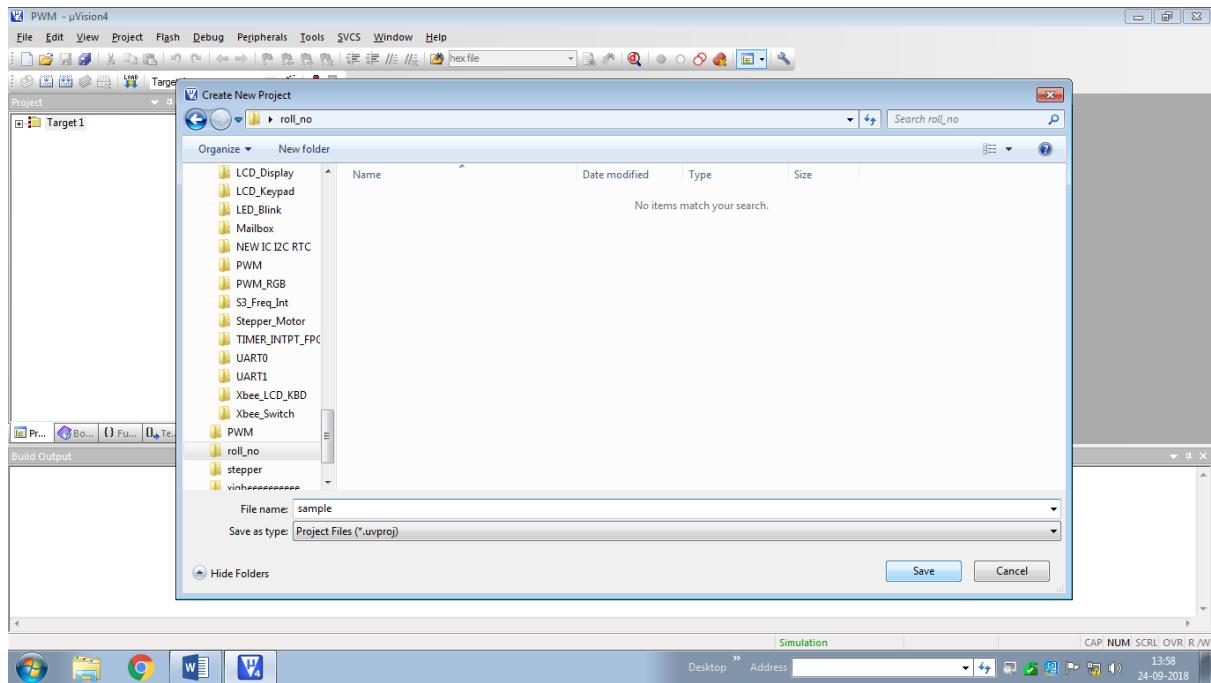
## 2. Click on Project and select New μVision Project.



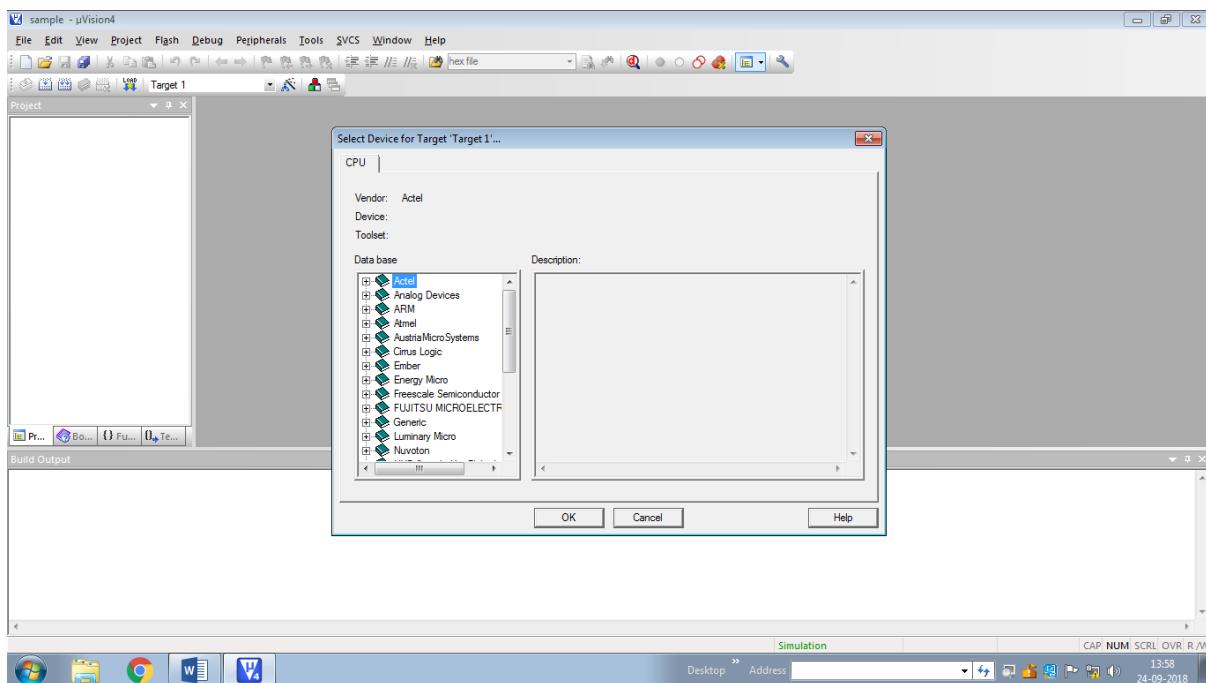
3. Create a new folder in Desktop. (during university practical use your register number as the folder name)



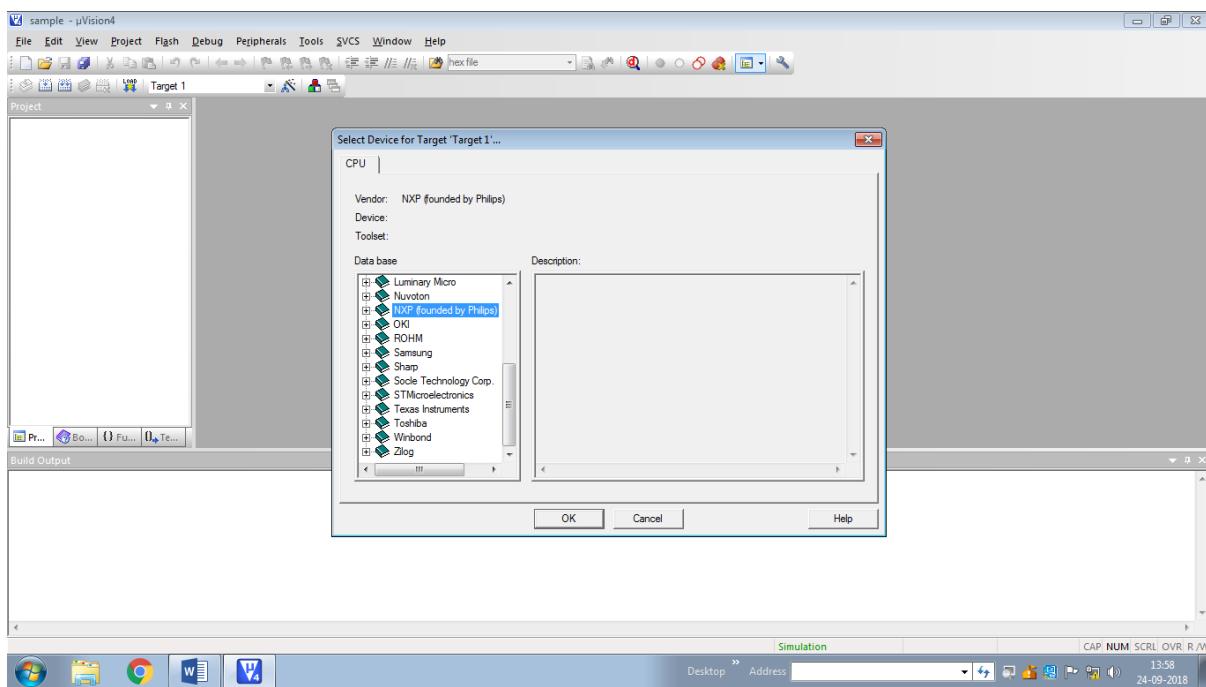
4. Type in the project name without any extension and save it.



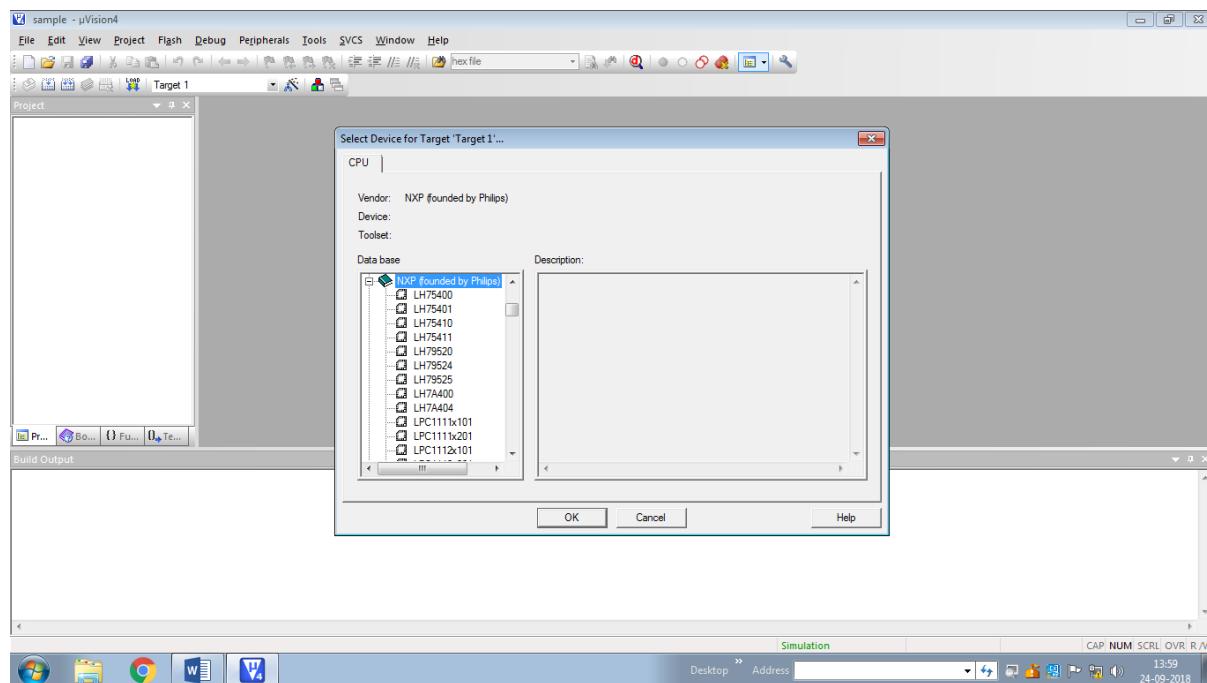
The following window will pop out



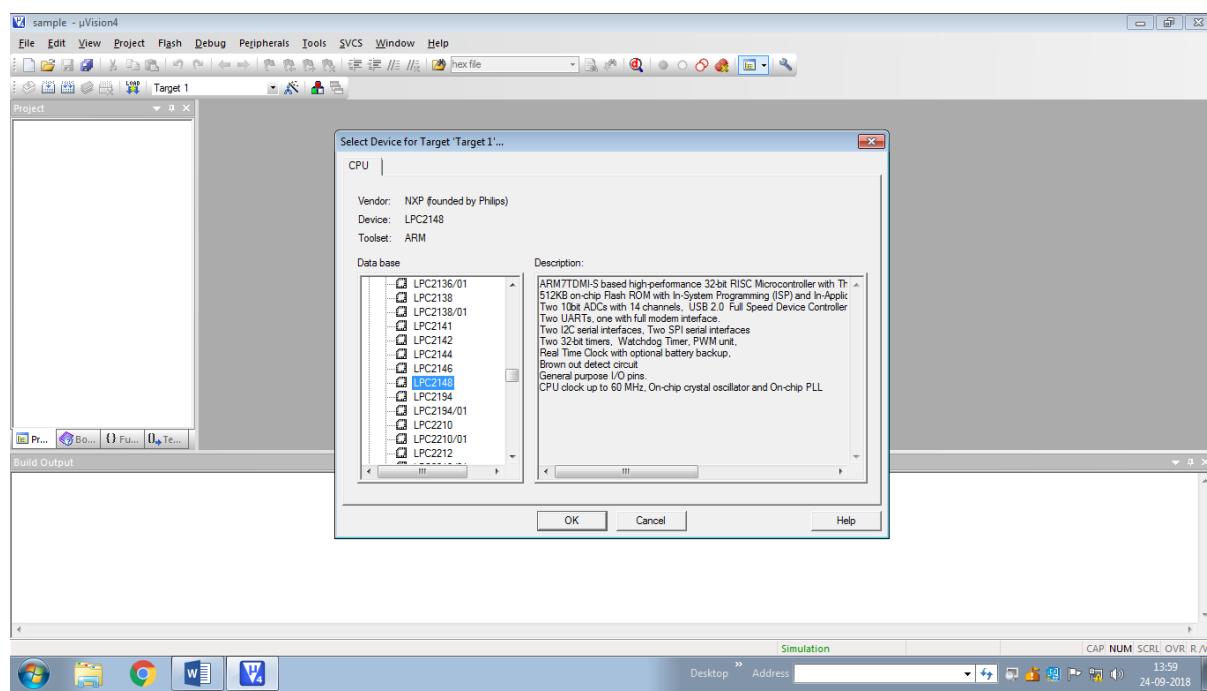
## 6. Select NXP (Founded by Philips) from Database window.



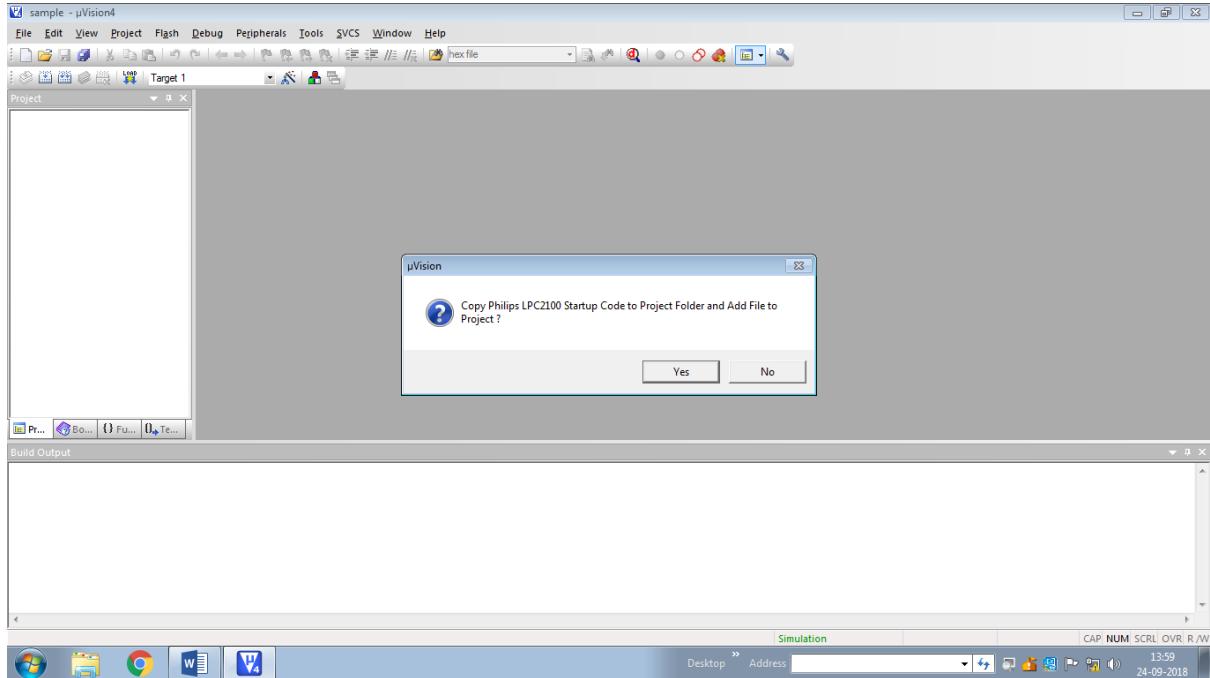
7. Click on the add (+) symbol present on the left of NXP.



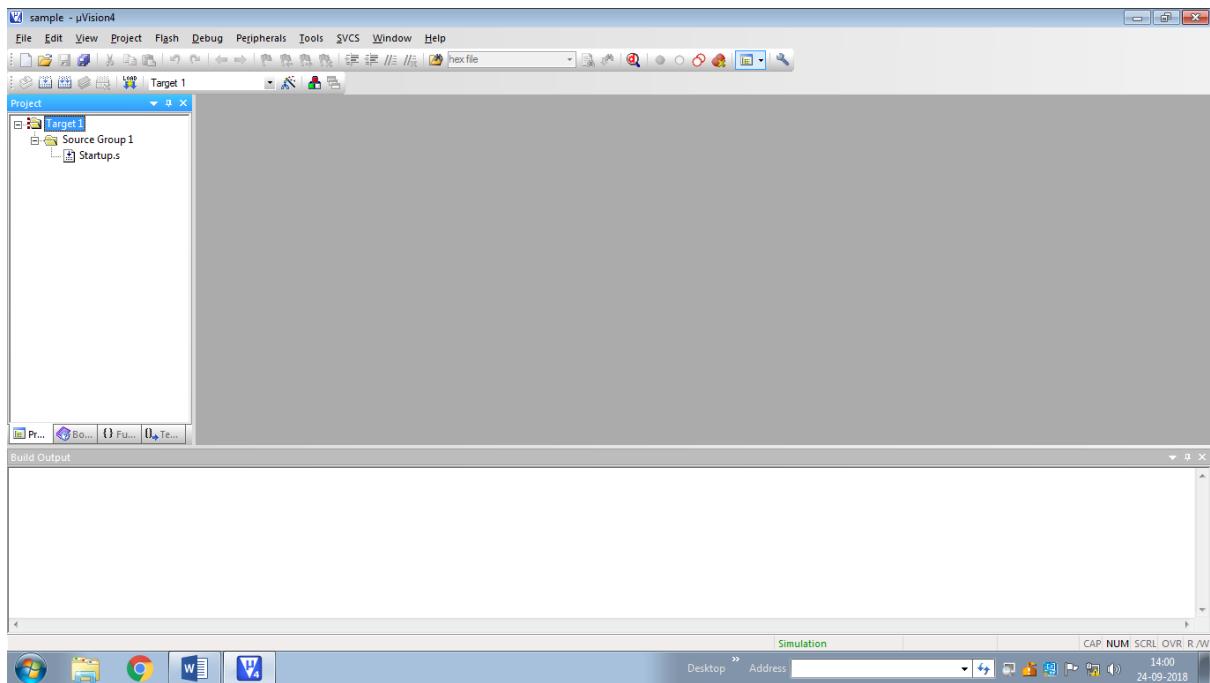
8. Select LPC2148 and give OK.



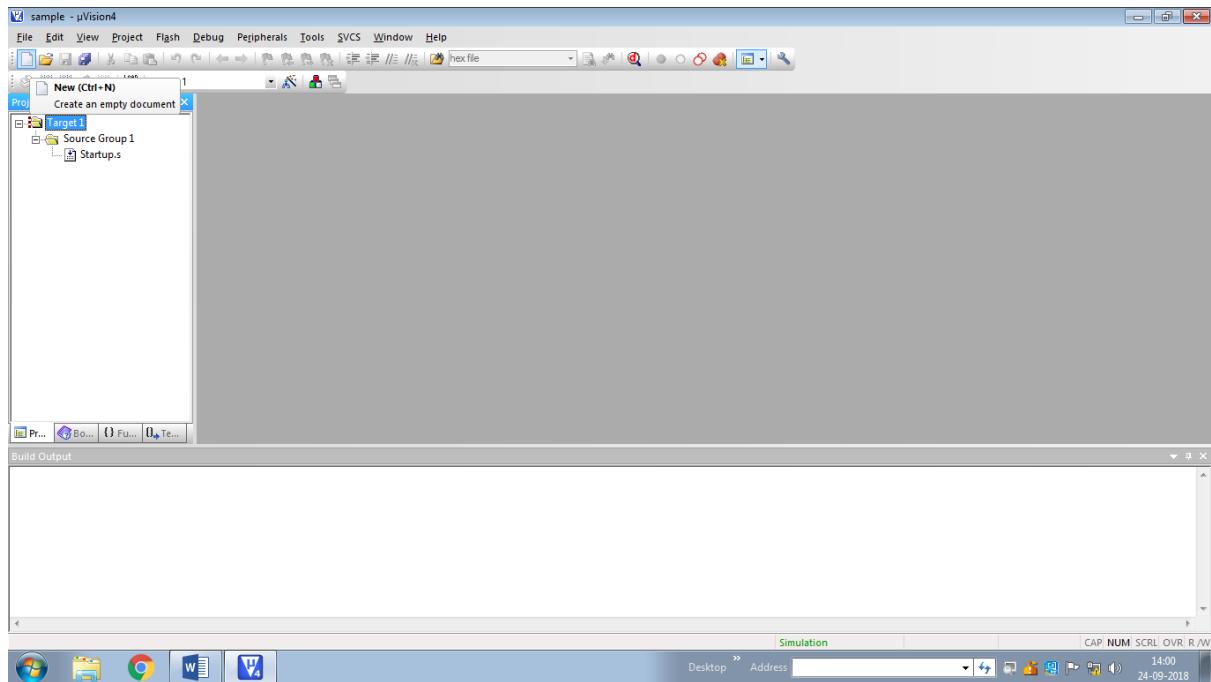
9. Select Yes.



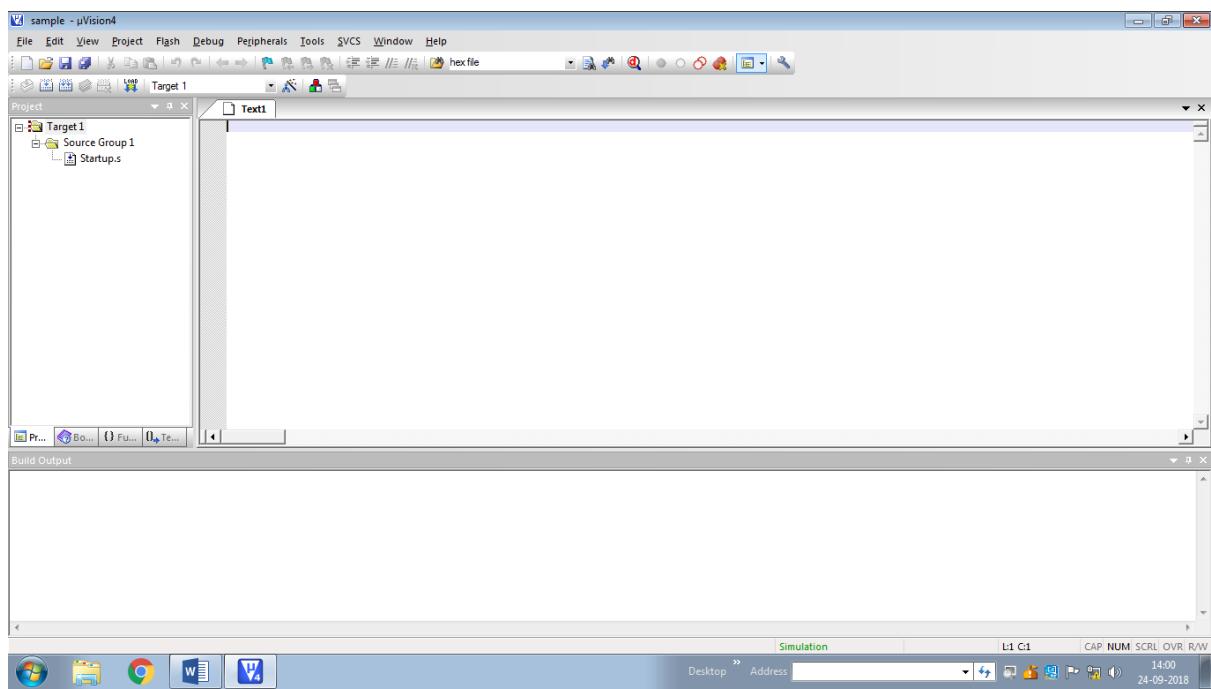
10. Source group is created in Target folder. (as shown in the Project window tab)



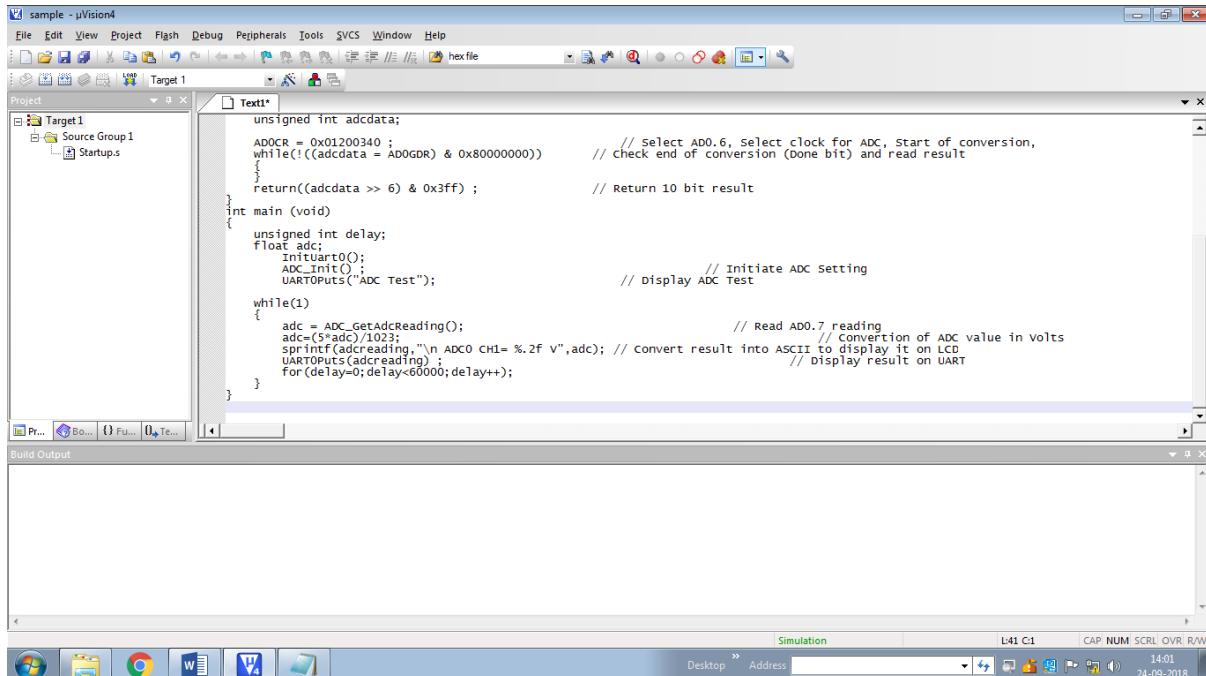
## 11. Create an Empty New document.



## 12. New document is created.

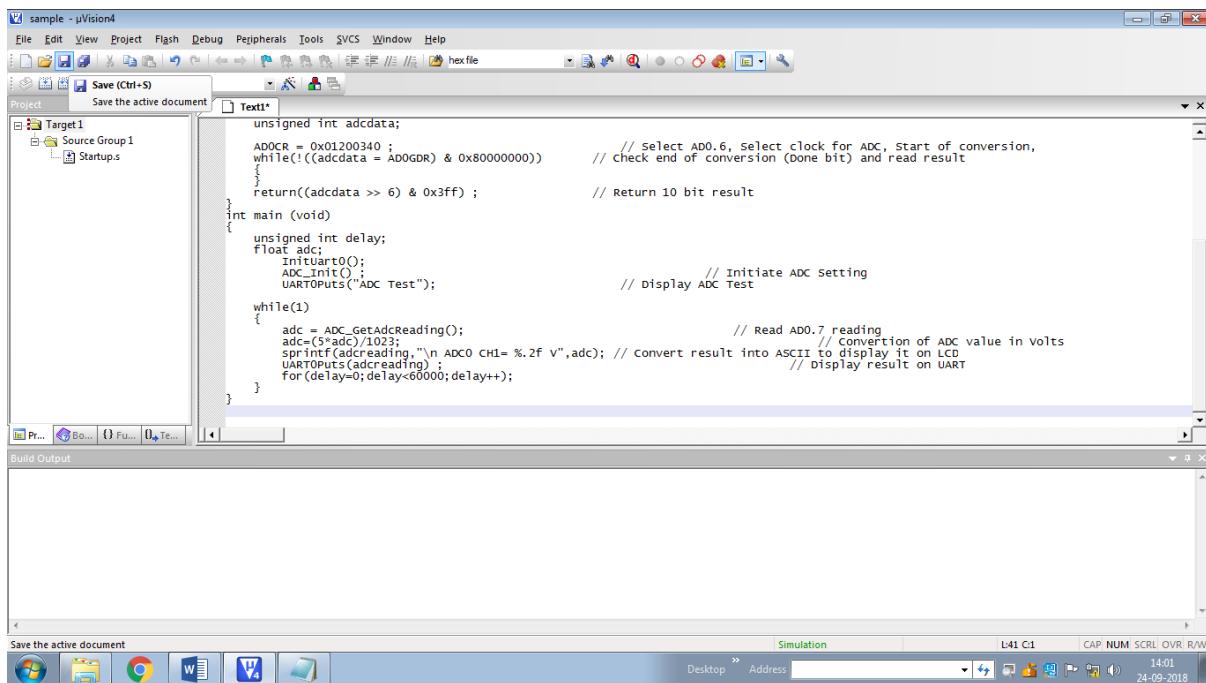


### 13. Type the program in the document.



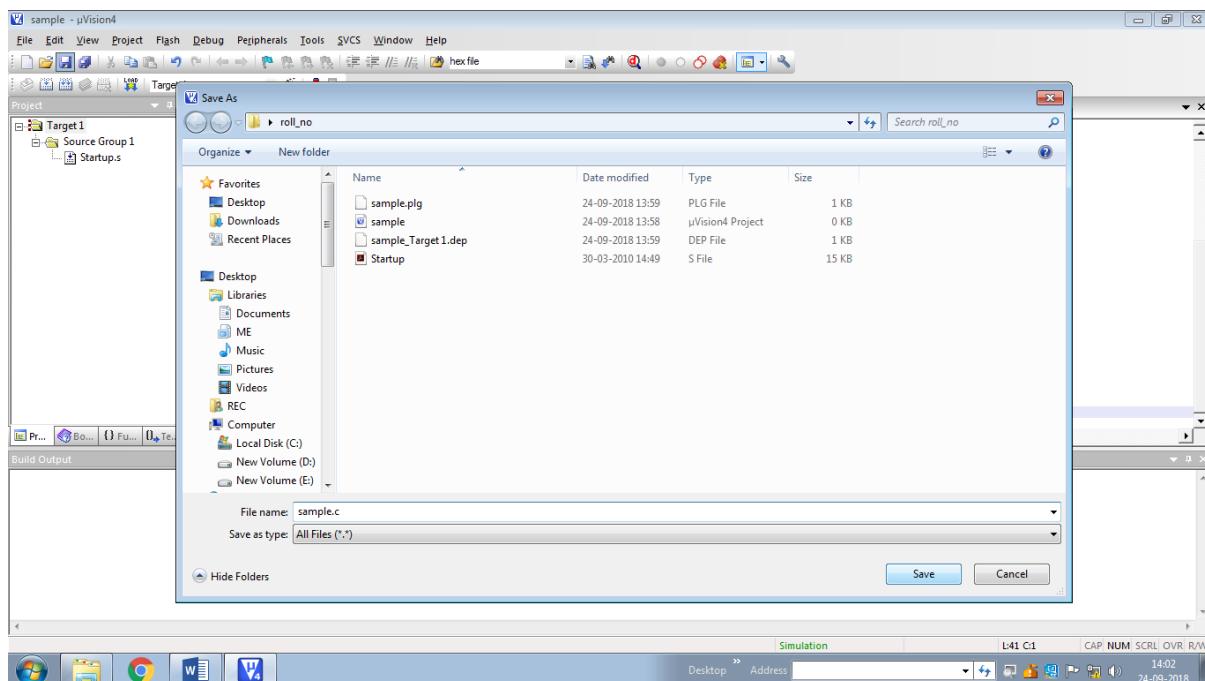
```
sample - uVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project Target 1 Source Group 1 Startup.s Text1
unsigned int adddata;
ADOCR = 0x01200340; // Select ADC0, Select clock for ADC, start of conversion,
while(!(adddata = AD0GDR) & 0x80000000); // check end of conversion (done bit) and read result
return((adddata >> 6) & 0x3ff); // Return 10 bit result
int main (void)
{
    unsigned int delay;
    float adc;
    inituart0();
    ADC_Init();
    UARTOPuts("ADC Test"); // Display ADC Test
    while(1)
    {
        adc = ADC_GetAdcReading(); // Read ADC0.7 reading
        adc=adc/1023; // Conversion of ADC value in volts
        sprintf(adcreading, "\n ADC0 CH1= %.2f V",adc); // convert result into ASCII to display it on LCD
        UARTOPuts(adcreading); // Display result on UART
        for(delay=0;delay<60000;delay++);
    }
}
Build Output
Simulation L41 C1 CAP NUM SCR LVR R/W
14.01 24-09-2018
```

### 14. Save the document.

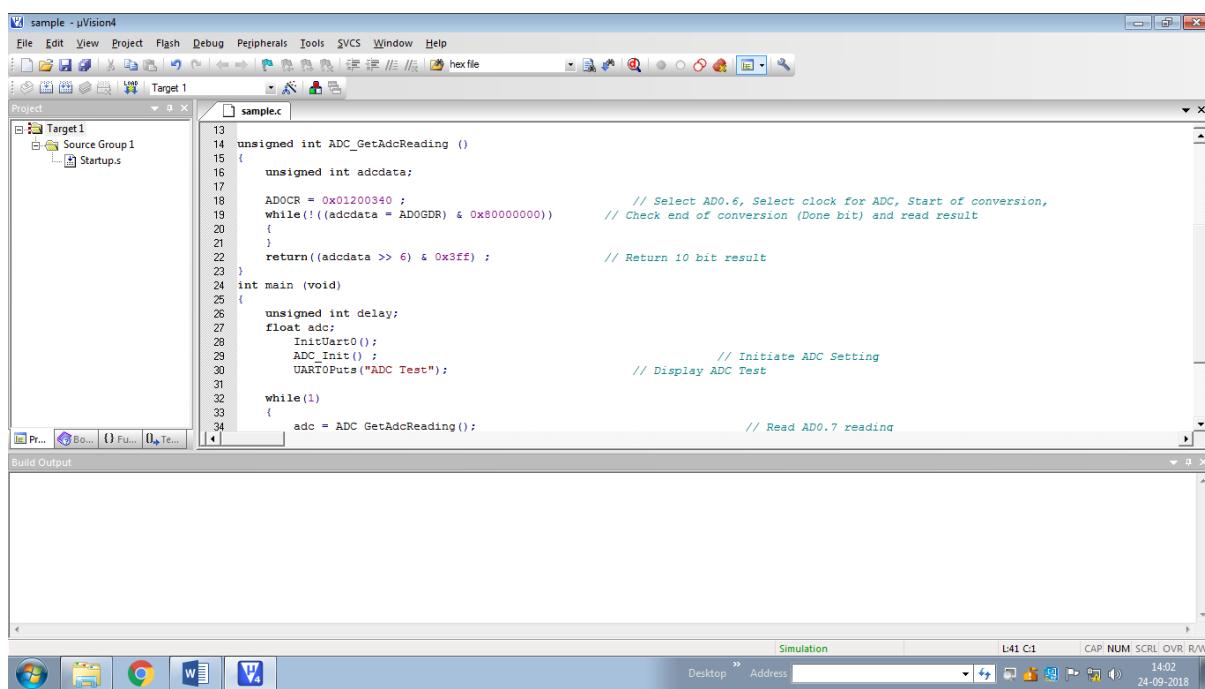


```
sample - uVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Save (Ctrl+S) Save the active document Project Target 1 Source Group 1 Startup.s Text1
unsigned int adddata;
ADOCR = 0x01200340; // Select ADC0, Select clock for ADC, start of conversion,
while(!(adddata = AD0GDR) & 0x80000000); // check end of conversion (done bit) and read result
return((adddata >> 6) & 0x3ff); // Return 10 bit result
int main (void)
{
    unsigned int delay;
    float adc;
    inituart0();
    ADC_Init();
    UARTOPuts("ADC Test"); // Display ADC Test
    while(1)
    {
        adc = ADC_GetAdcReading(); // Read ADC0.7 reading
        adc=adc/1023; // Conversion of ADC value in volts
        sprintf(adcreading, "\n ADC0 CH1= %.2f V",adc); // convert result into ASCII to display it on LCD
        UARTOPuts(adcreading); // Display result on UART
        for(delay=0;delay<60000;delay++);
    }
}
Build Output
Simulation L41 C1 CAP NUM SCR LVR R/W
14.01 24-09-2018
```

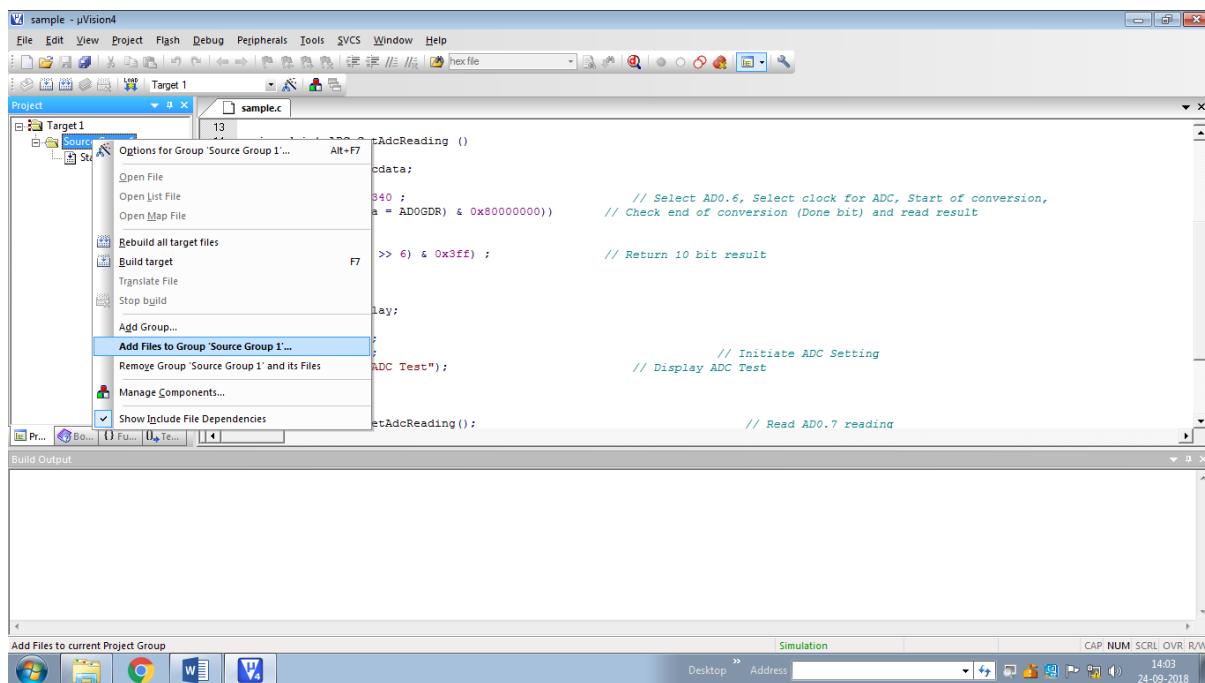
15. Save the document with extension .c inside the project folder.



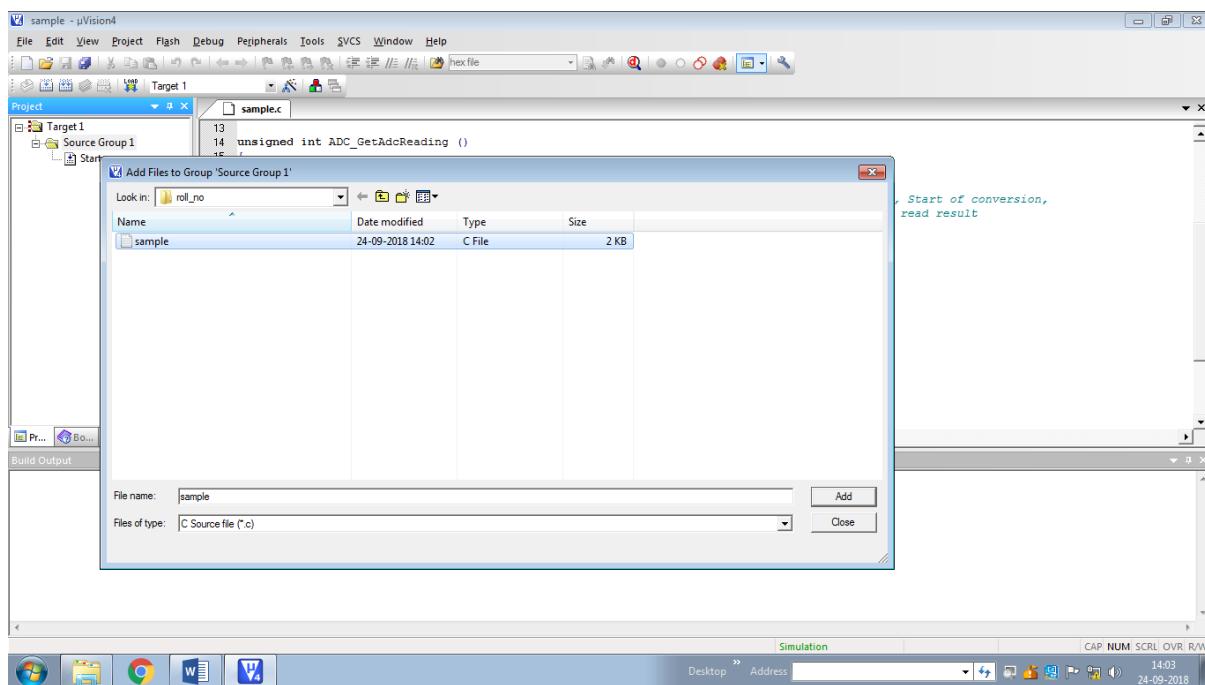
You can see the change of colour of the text in the saved file



17. Right click on source group and select add files to group ‘Source group 1’.



18. Add the **sample.c** file which is already created and click Add option.



19. The sample.c file will be visible under the Source group 1.

The screenshot shows the µVision4 IDE interface. The menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains icons for file operations like Open, Save, and Build. The project tree on the left shows Target 1 with Source Group 1 containing Startups and sample.c. The main editor window displays the following C code:

```
sample - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
[...] hex file
Project Target 1
sample.c
13
14     unsigned int ADC_GetAdcReading () {
15     {
16         unsigned int adcdatas;
17
18         ADOCR = 0x01200340 ;           // Select ADO.6, Select clock for ADC, Start of conversion,
19         while (!(adcdatas = ADOGDR) & 0x80000000)    // Check end of conversion (Done bit) and read result
20         {
21         }
22         return ((adcdatas >> 6) & 0x3fff) ;          // Return 10 bit result
23     }
24     int main (void)
25     {
26         unsigned int delay;
27         float adc;
28         Iniruart0();
29         ADC_Init() ;
30         UARTOPuts("ADC Test");                  // Initiate ADC Setting
31
32         while(1)
33         {
34             adc = ADC_GetAdcReading();           // Read ADO.7 reading

```

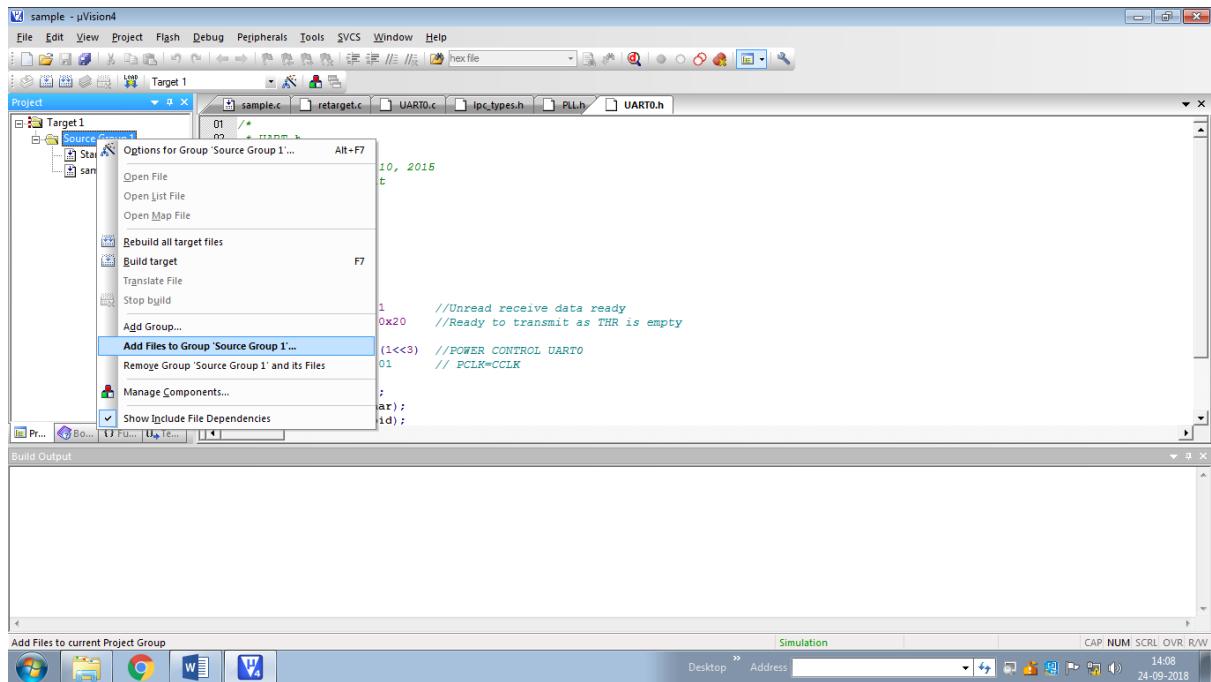
20. Create the c and h files needed to compile your project using the above steps as in 12, 13 and 14.

The screenshot shows the µVision4 IDE interface with the following details:

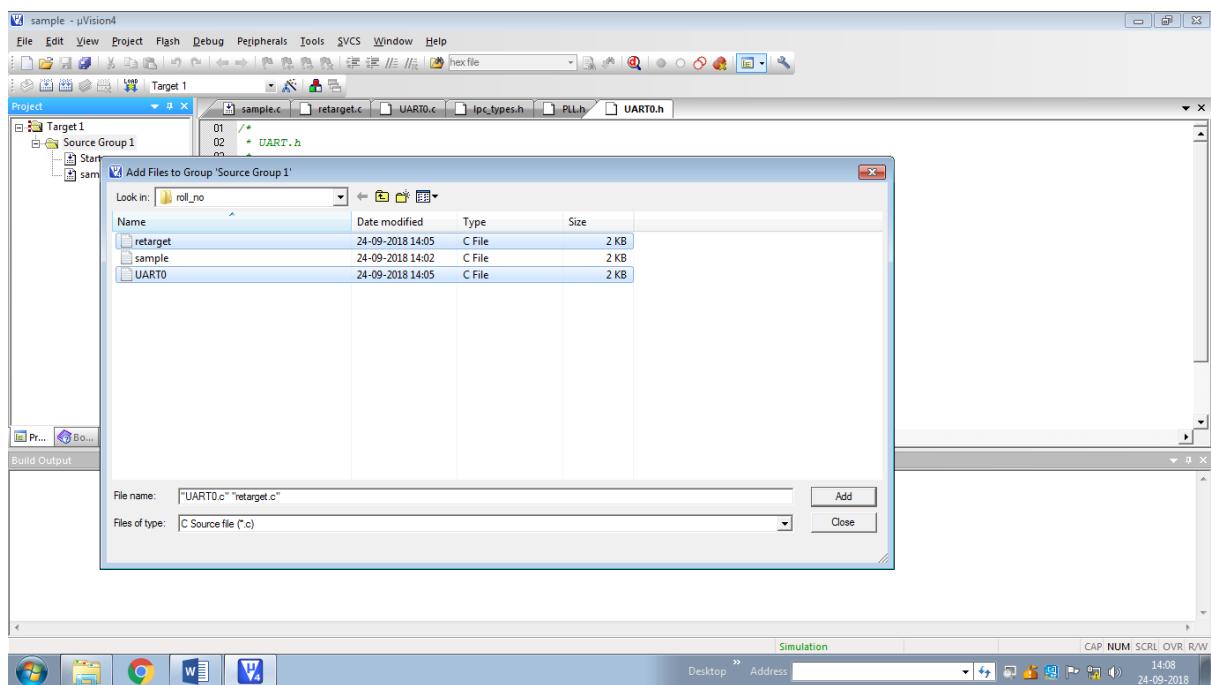
- Title Bar:** sample - µVision4
- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help
- Toolbar:** Includes icons for Open, Save, Build, Run, and Simulation.
- Project Explorer:** Shows Target1 with Source Group 1 containing Startup.s and sample.c.
- Code Editor:** The current file is UART0.h, displaying the following code:

```
01 /*+  
02 *  UART.h  
03 *+  
04 *  Created on: Jul 10, 2015  
05 *  Author: Ankit  
06 */  
07  
08 #ifndef UART0_H_  
09 #define UART0_H_  
10  
11  
12 #include <stdio.h>  
13  
14 #define LSR_RDR      1      //Unread receive data ready  
15 #define LSR_THRE    0x20    //Ready to transmit as THR is empty  
16  
17 #define PCLKUART0   (1<<3) //POWER CONTROL UART0  
18 #define PCLK_UART0  01      // PCLK=CCLK  
19  
20 void InitUart0(void);  
21 char UART0putchar(char);  
22 char UART0getchar(void);
```
- Build Output:** A panel at the bottom left showing build status.
- Bottom Bar:** Includes icons for Windows Start, Task View, Google Chrome, and a workspace switcher. The workspace switcher shows a W4 icon.
- Status Bar:** Simulation, L26 C1, CAP NUM SCR L/R/W, and a date/time indicator (14:07, 24-09-2018).

21. Click on Add files to Group ‘Source group 1’ under the Source group 1.



22. Select the newly created .c files and click Add option.



23. C files will be added to the Source group 1.

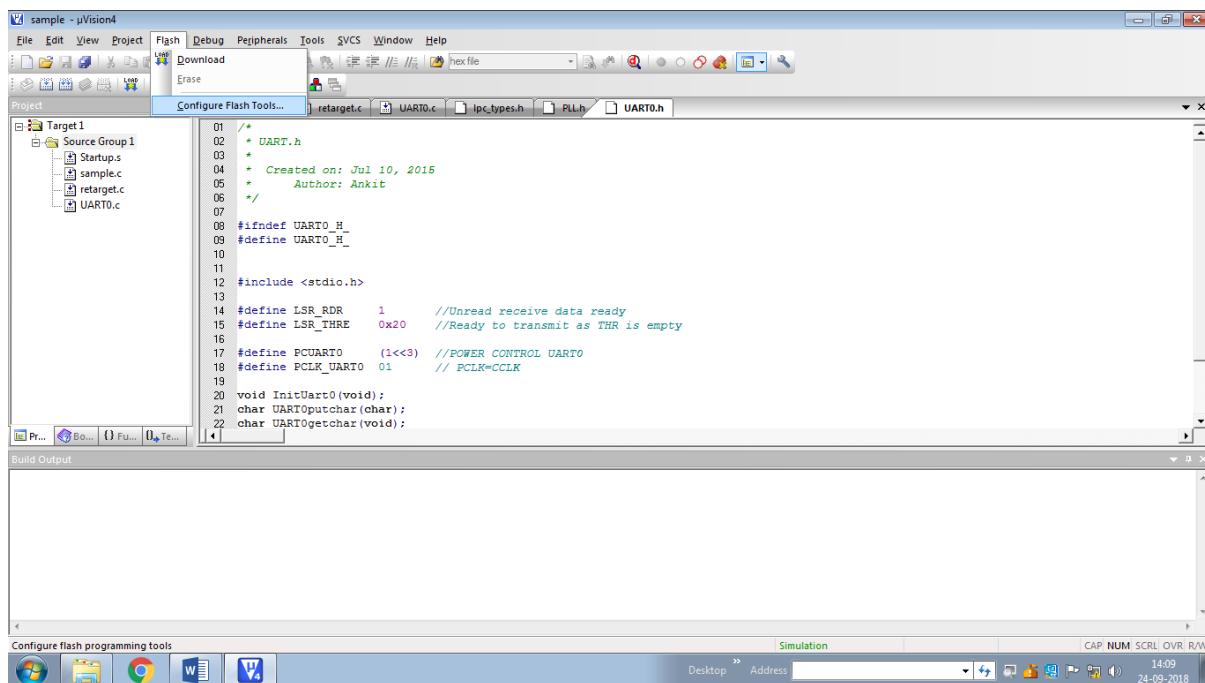
The screenshot shows the Keil uVision4 IDE interface. The top menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations like Open, Save, and Build. The main window has tabs for sample.c, retarget.c, UART0.c, ipc\_types.h, PLL.h, and UART0.h. The Project pane on the left shows Target1 with Source Group 1 containing Startup.s, sample.c, retarget.c, and UART0.c. The code editor pane displays the content of UART0.c, which includes header definitions, include statements, and function prototypes. Below the code editor is a Build Output pane. The bottom status bar shows the date and time as 24-09-2018.

```
01 /*  
02 *  UART.h  
03 *  
04 *  Created on: Jul 10, 2015  
05 *  Author: Ankit  
06 */  
07  
08 #ifndef UART0_H_  
09 #define UART0_H_  
10  
11  
12 #include <stdio.h>  
13  
14 #define LSR_RDR    1      //Unread receive data ready  
15 #define LSR_THRE   0x20    //Ready to transmit as THR is empty  
16  
17 #define PCUART0  (1<<3) //POWER CONTROL UART0  
18 #define PCLK_UART0 01     // PCLK=CCLK  
19  
20 void InitUart0(void);  
21 char UART0putchar(char);  
22 char UART0getchar(void);
```

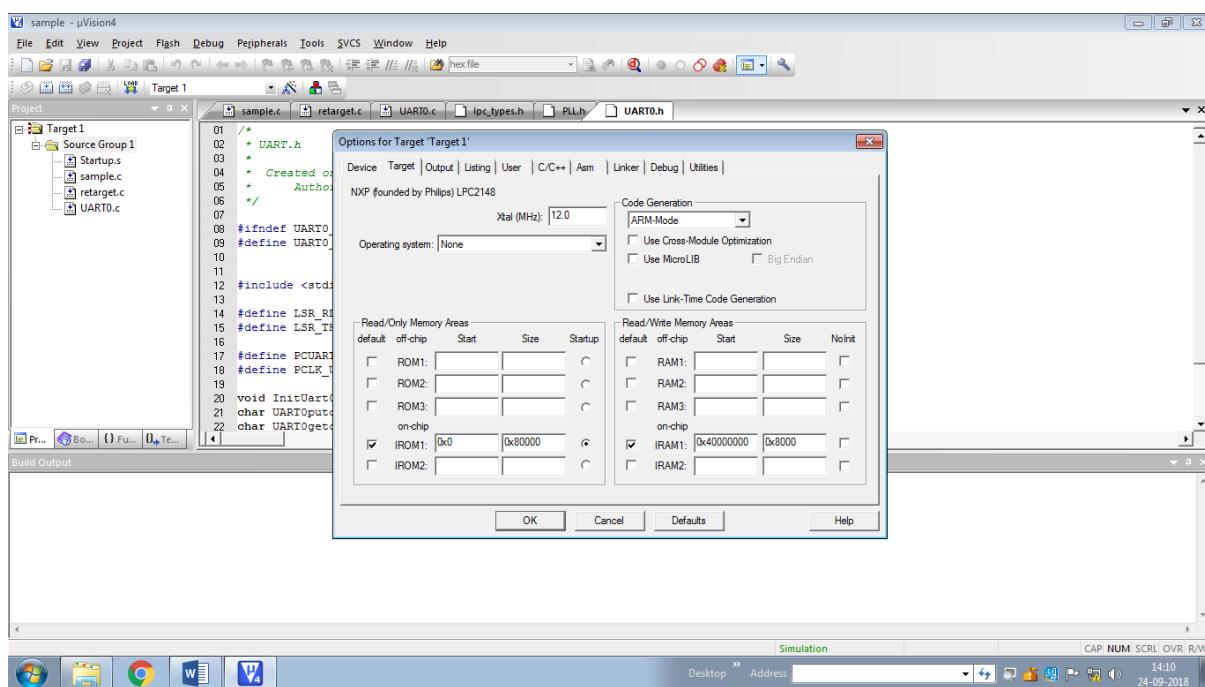
24. Click on save all.

This screenshot is similar to the previous one but with a notable difference: the 'Save All' icon in the toolbar is highlighted, indicating it has been clicked. The rest of the interface, including the project structure and code editor, remains the same.

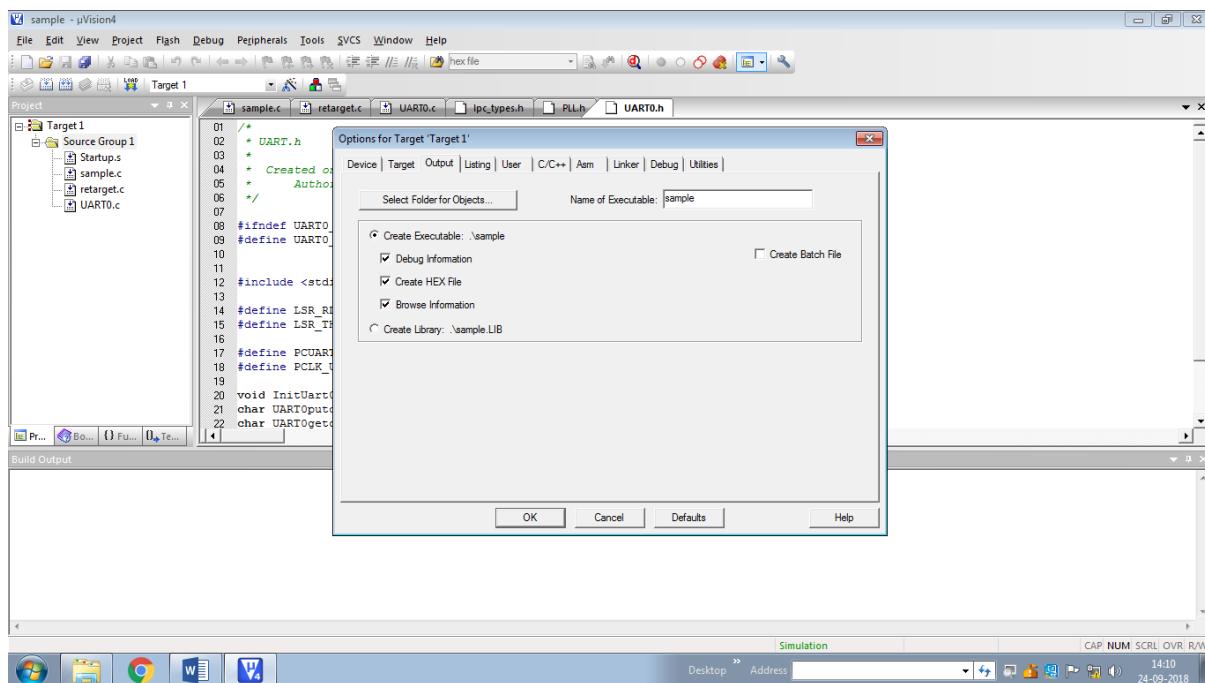
25. Click on Flash and select Configure Flash Tools.



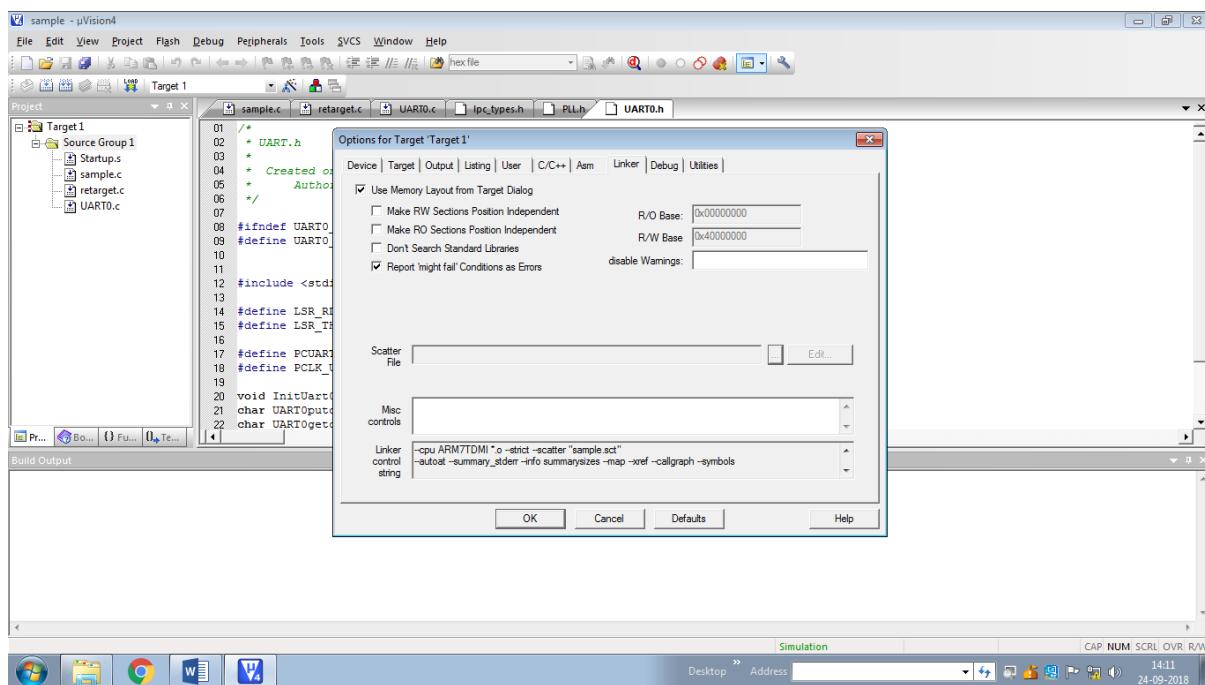
26. In the Target tab select the Xtal as 12 MHz.



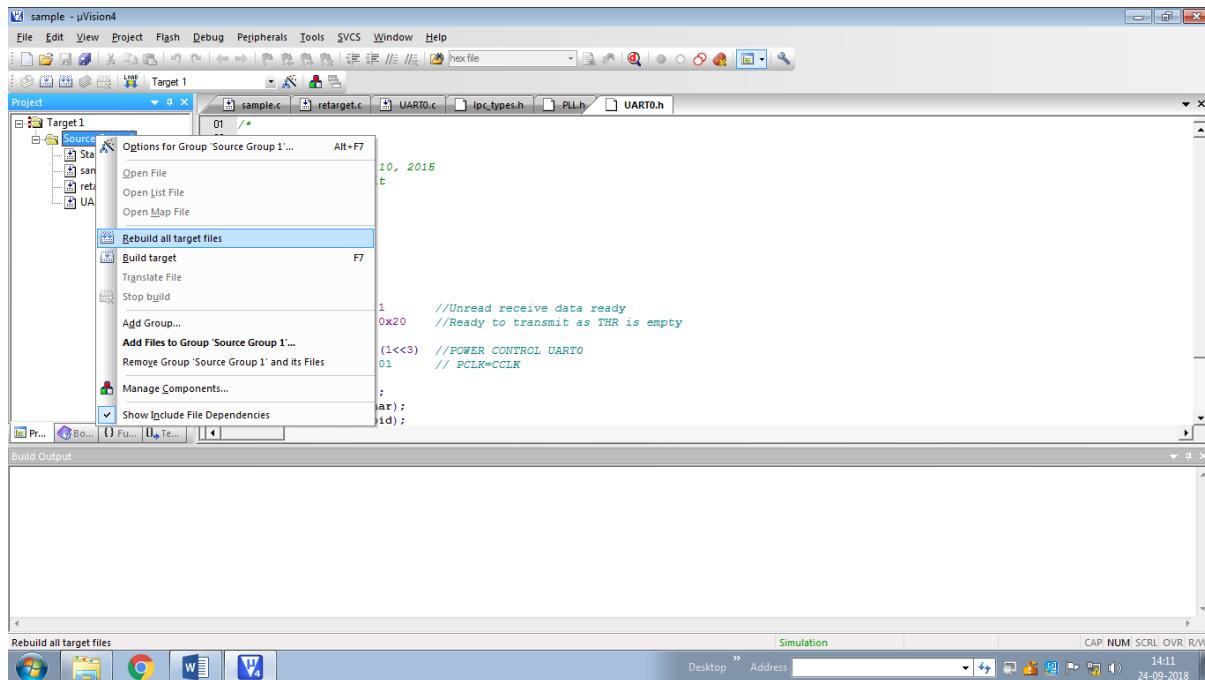
27. In the output tab select Create HEX file.



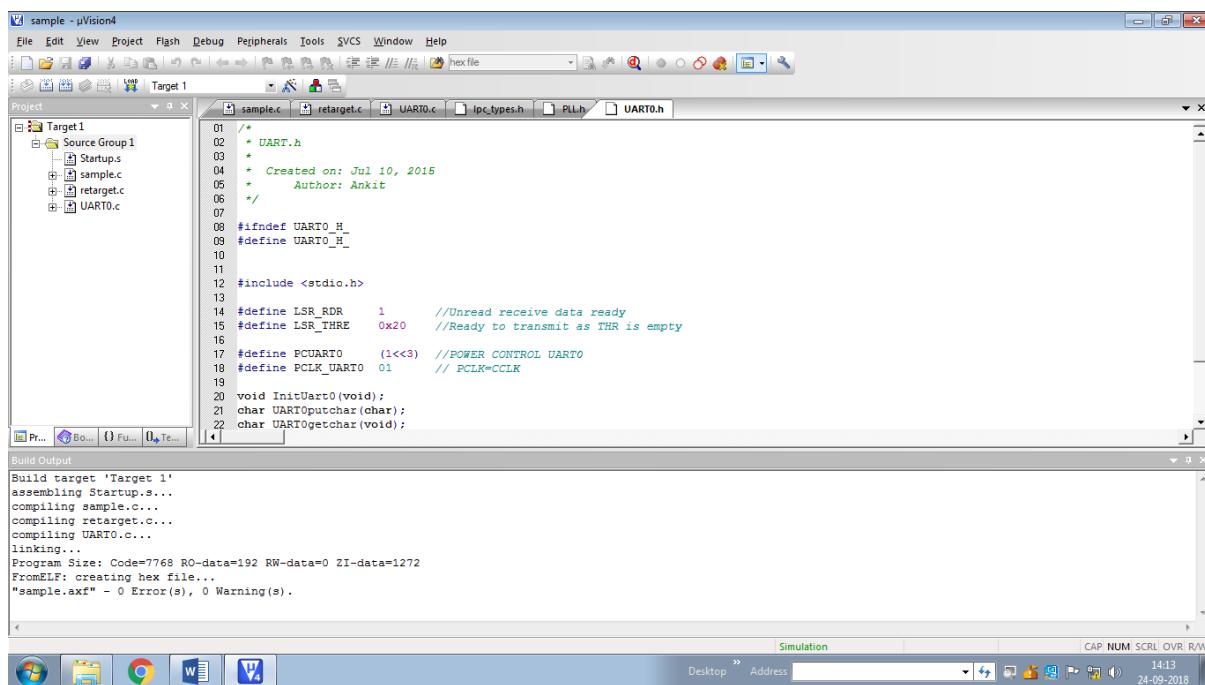
28. In Linker tab select Use memory layout from target dialog.



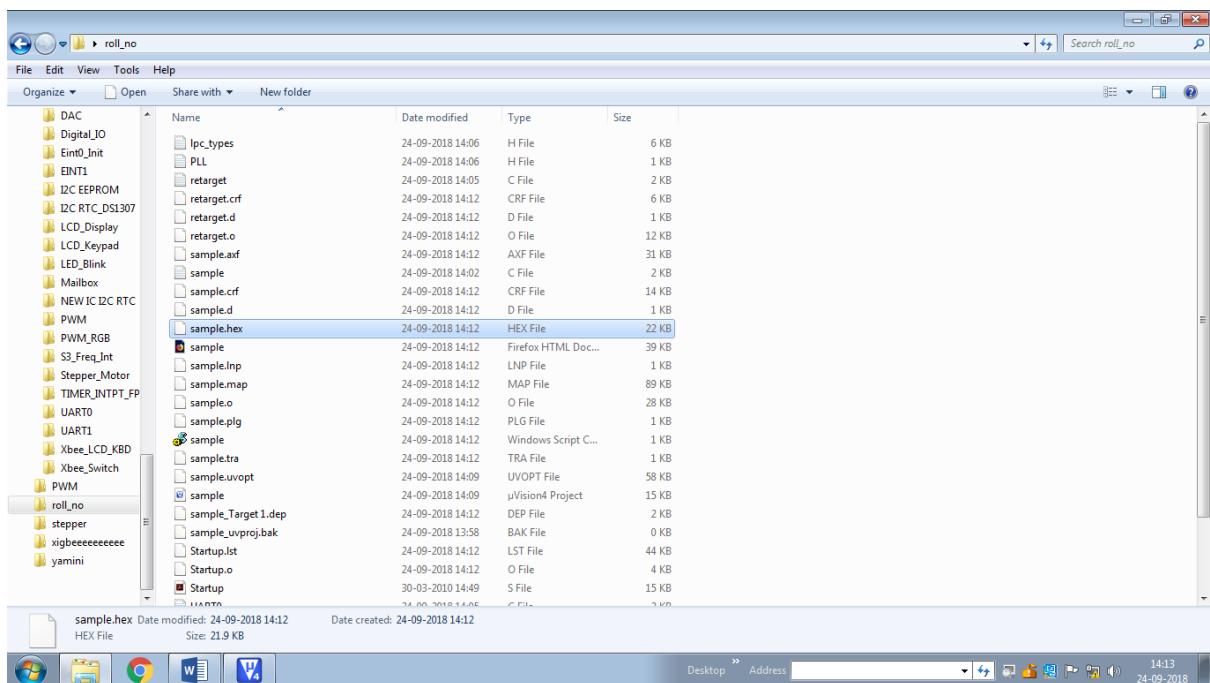
29. Right click on source group and select rebuild all target files.



30. HEX file will be created (else check for the errors and warnings)



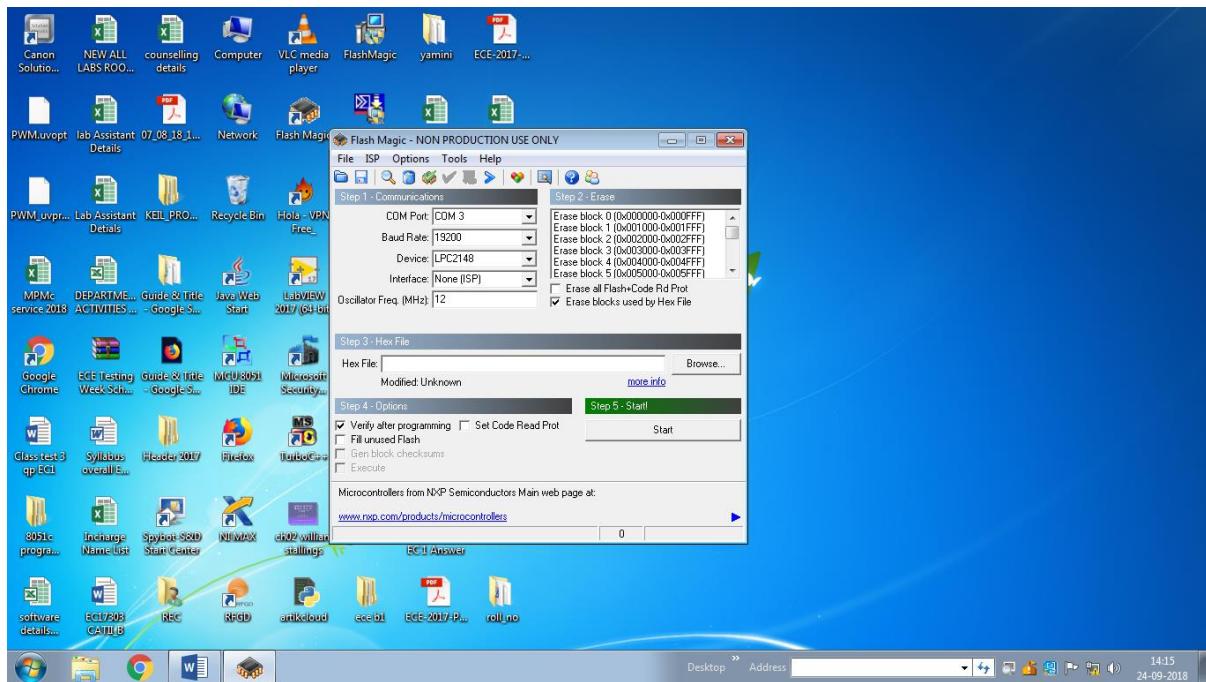
31. Hex file will be created in the folder.



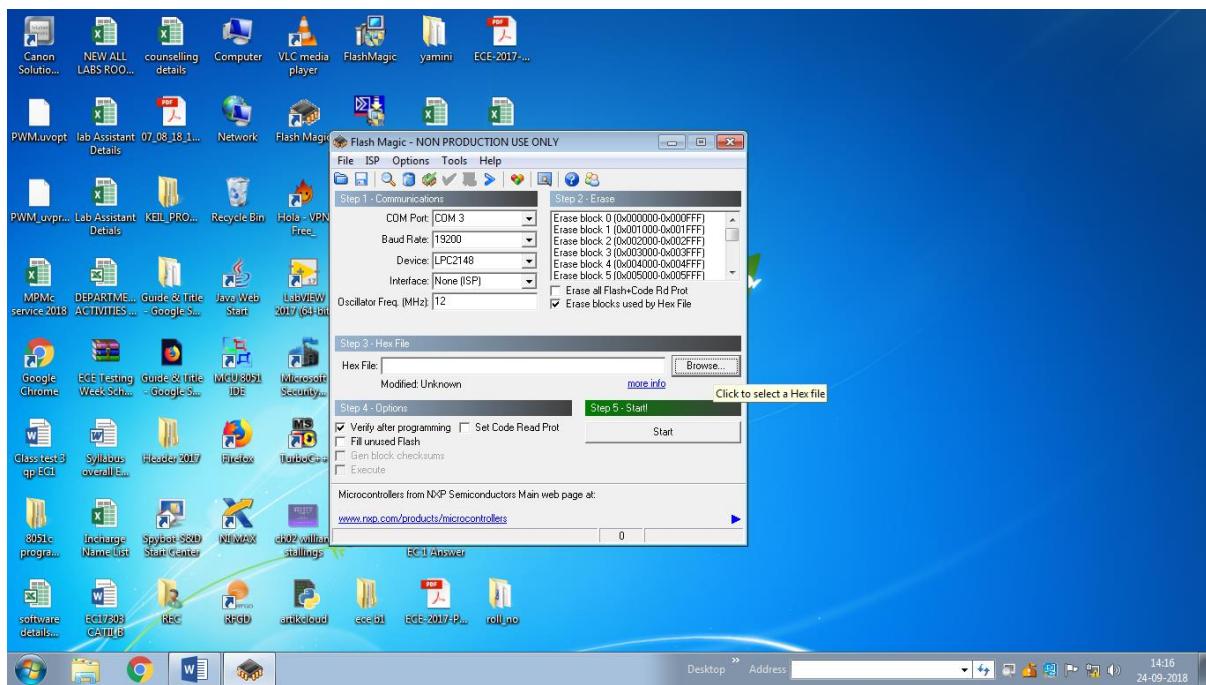
32. Select Flash Magic from the start menu.



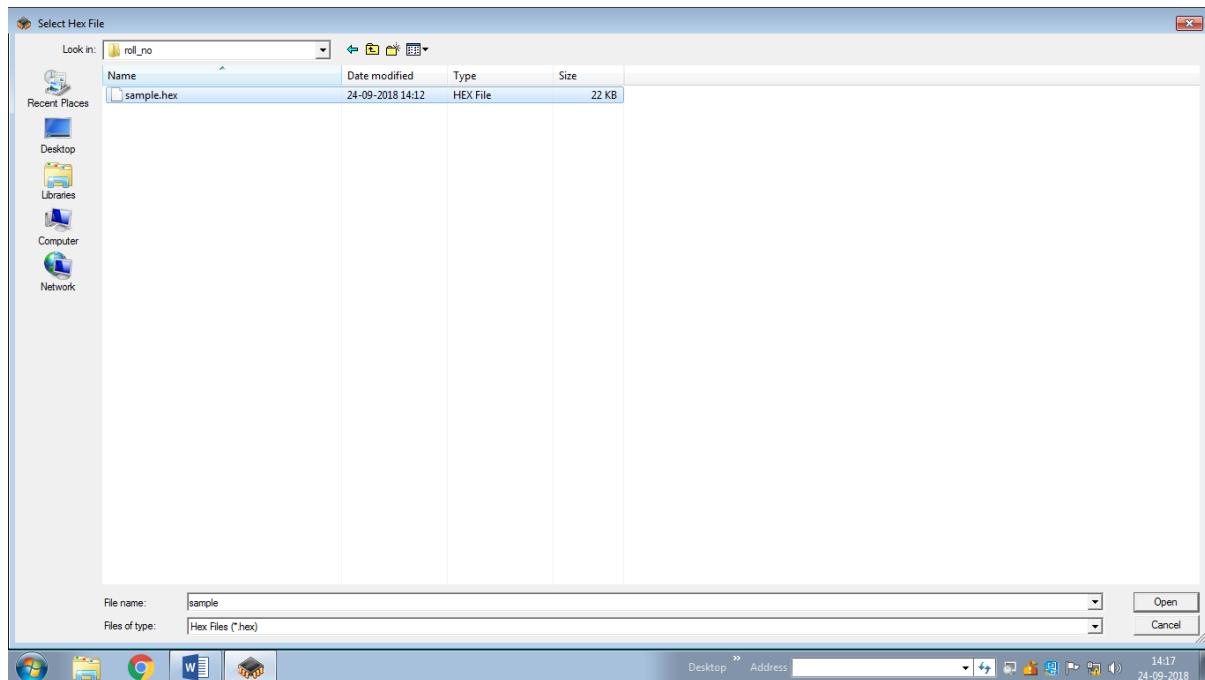
33. Make sure the following settings are there on the flash magic tool Step 1 tab.



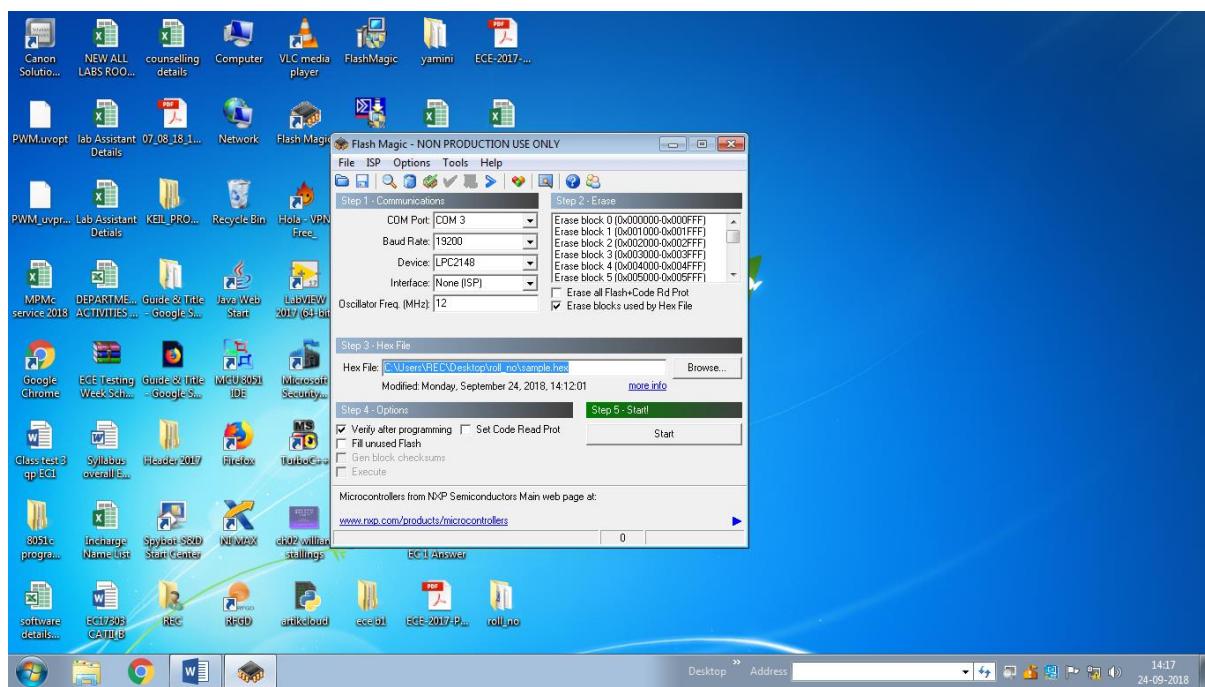
34. Click on the browse option.



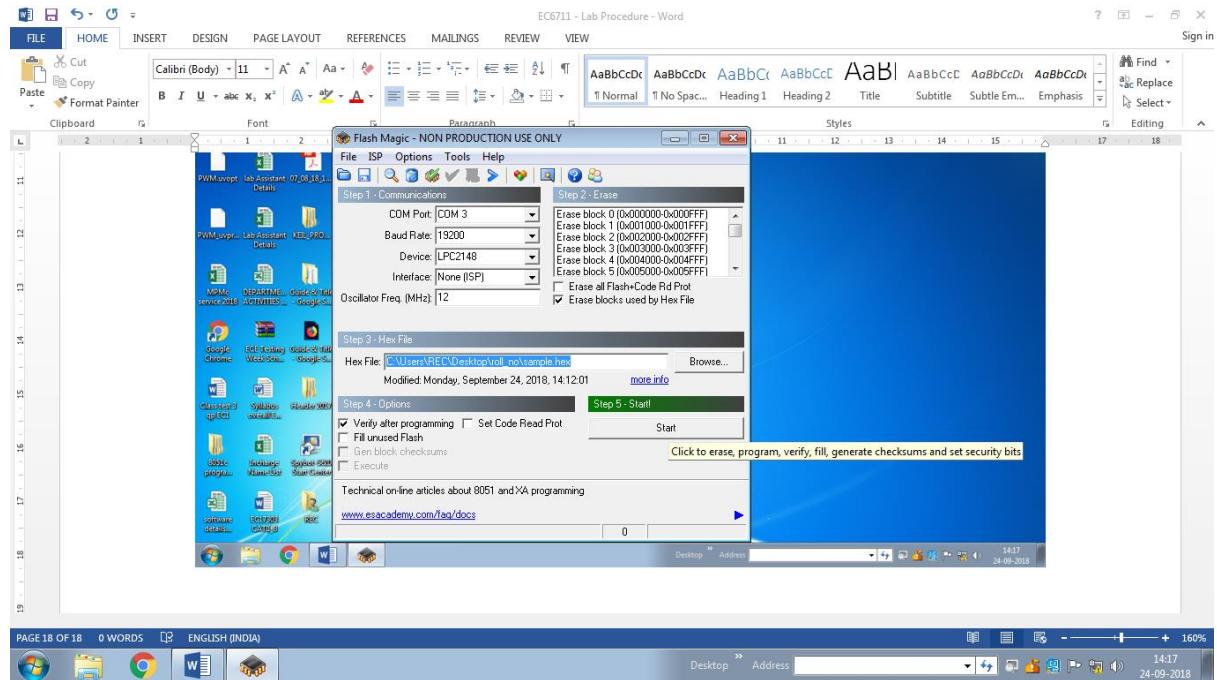
35. Open the HEX file created (from the respective project folder)



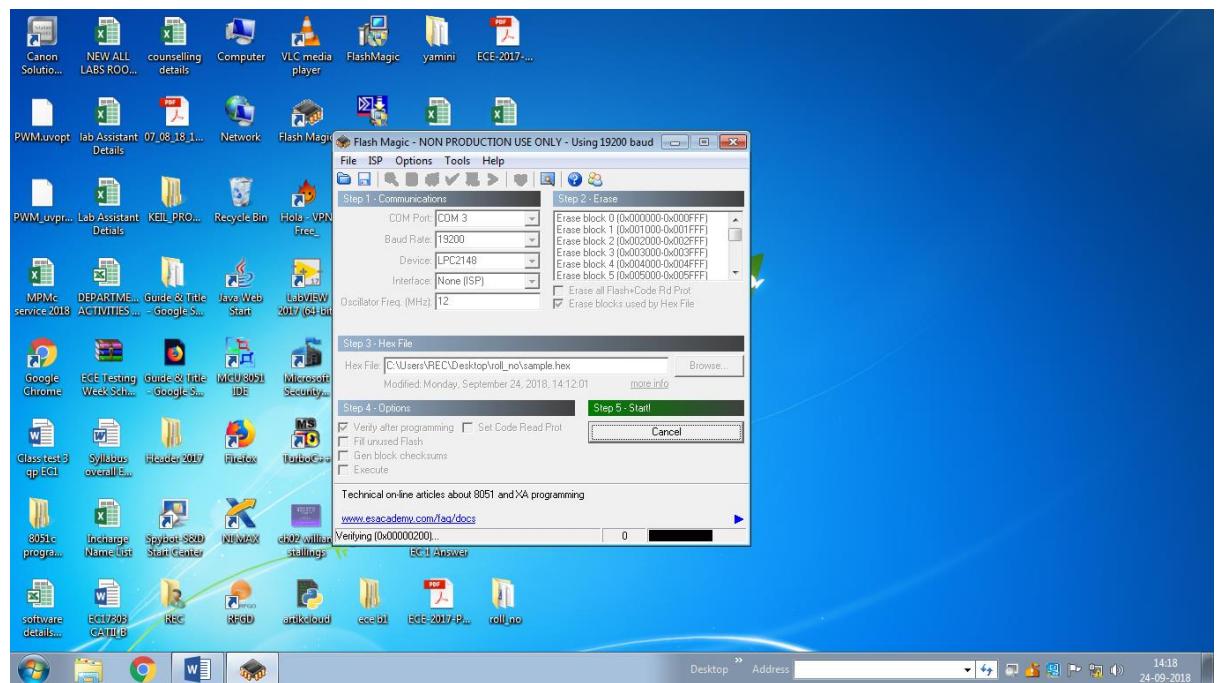
36. HEX file path name will be displayed.



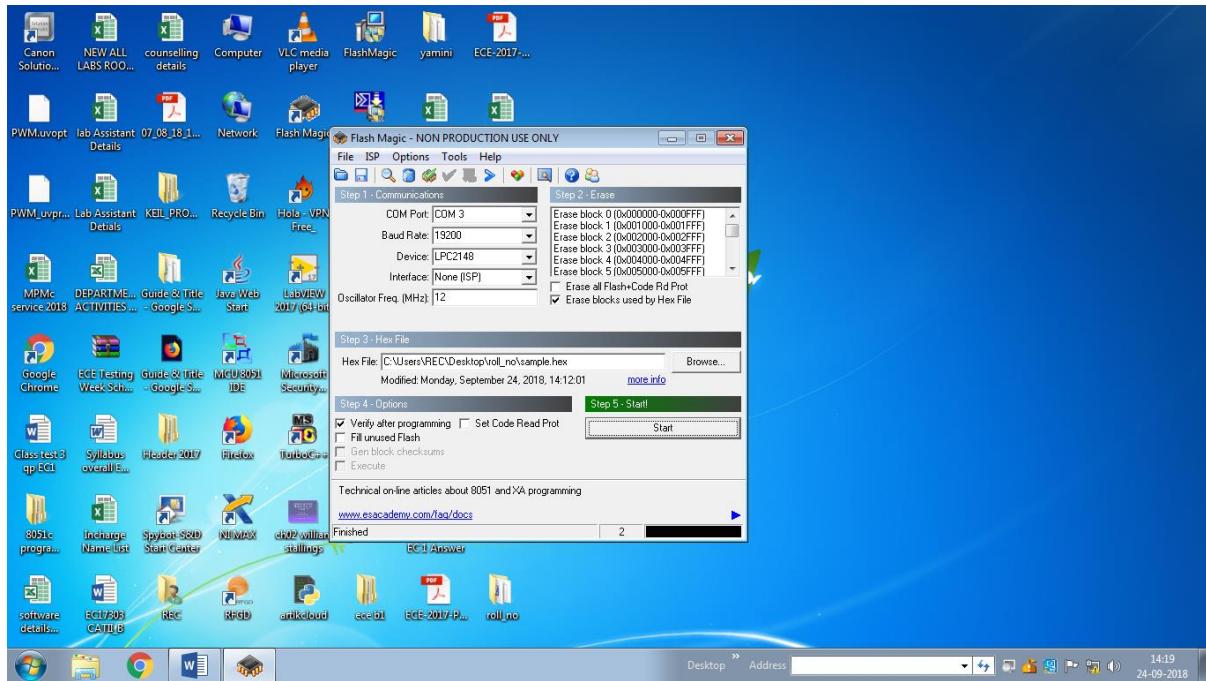
37. Click on start option to transfer the HEX file on to the kit.



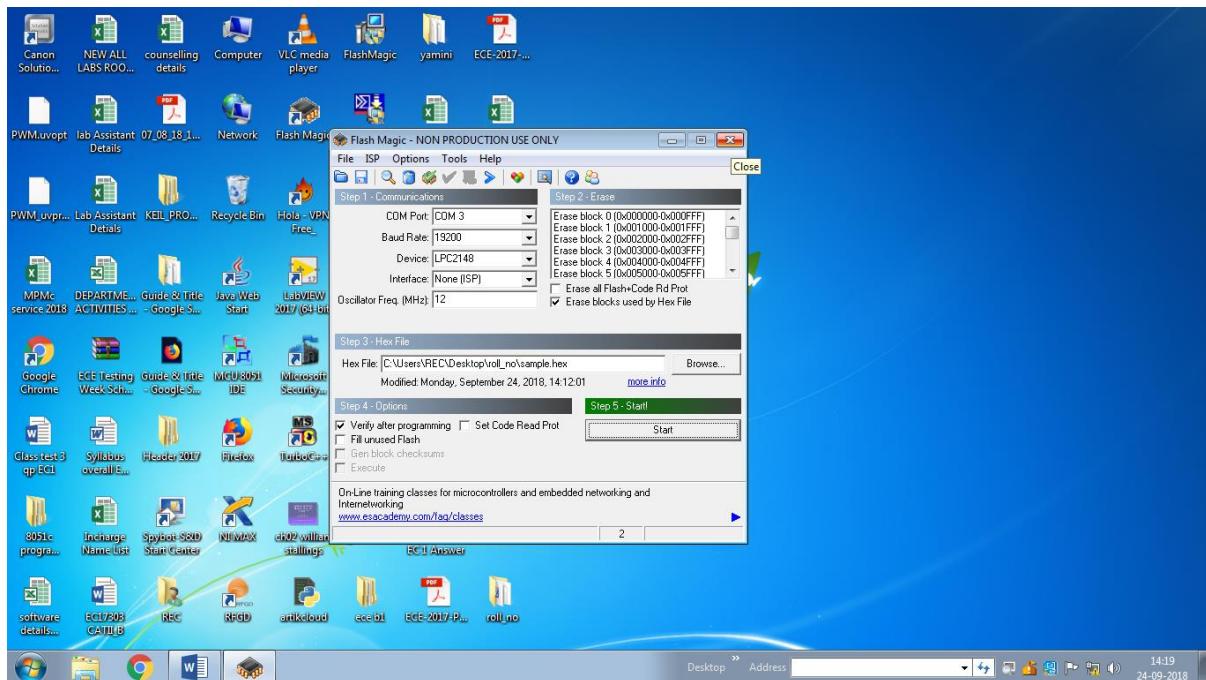
38. The HEX file will be verified and loaded.



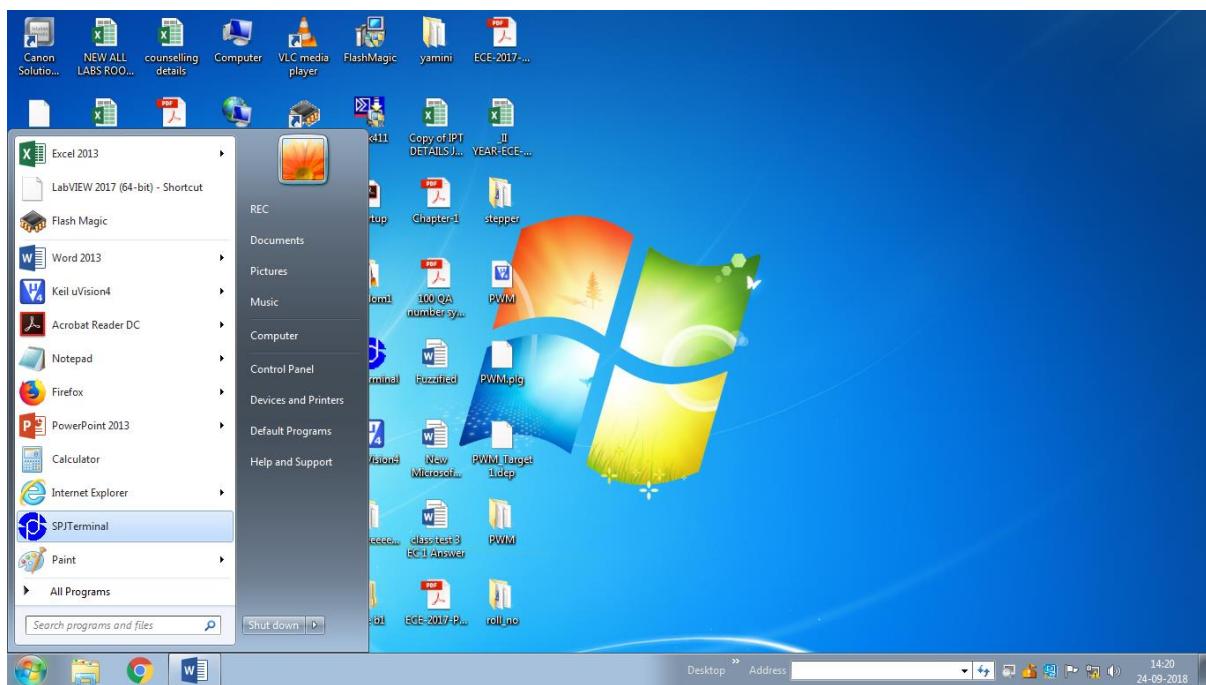
39. Loading is finished (indicated at the left bottom corner of the flash magic window)



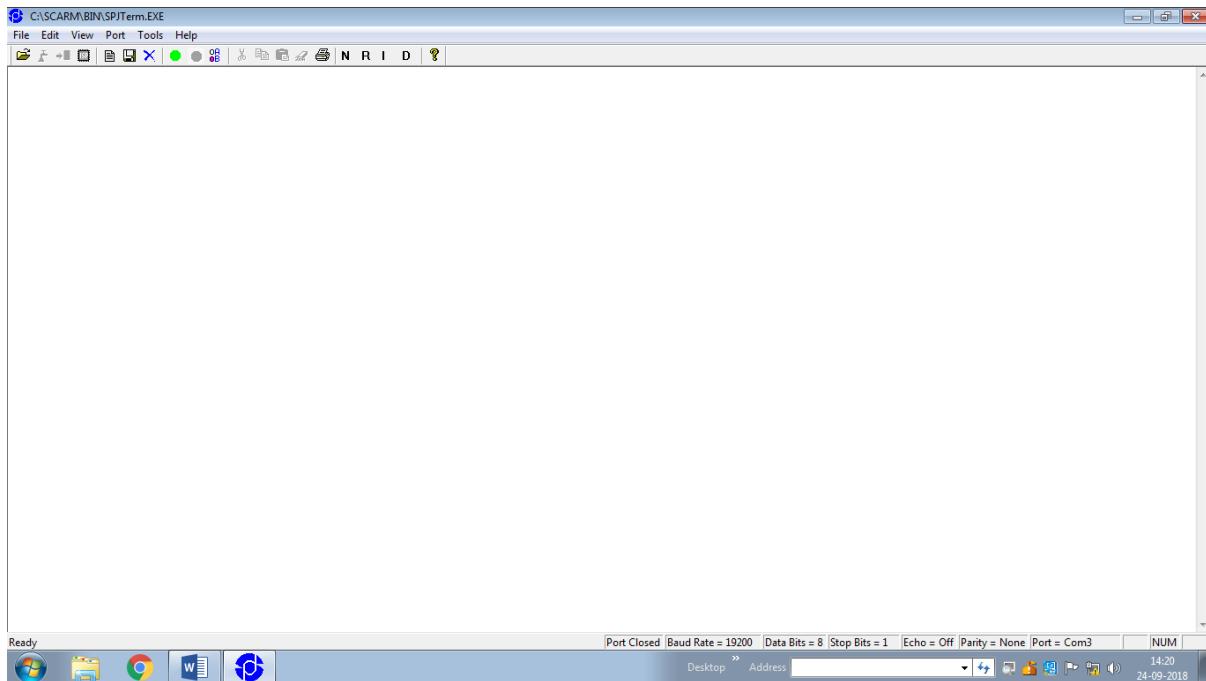
40. Close the Flash Magic window.



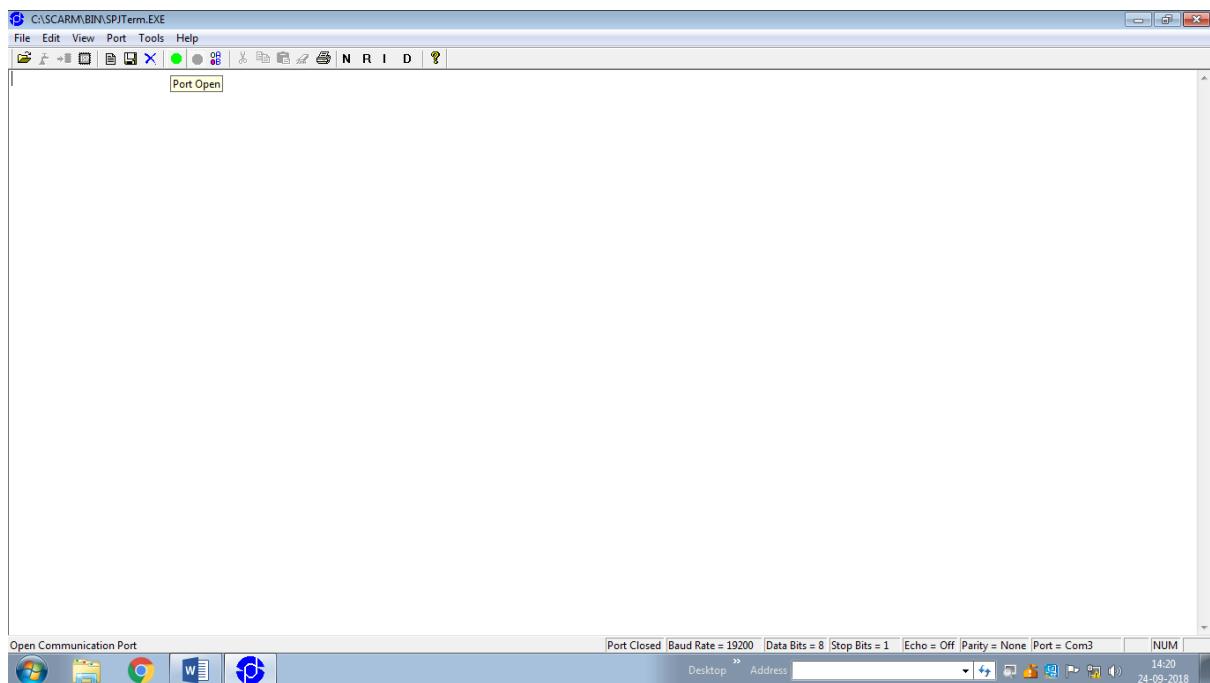
41. Select the SPJTerminal from start menu.



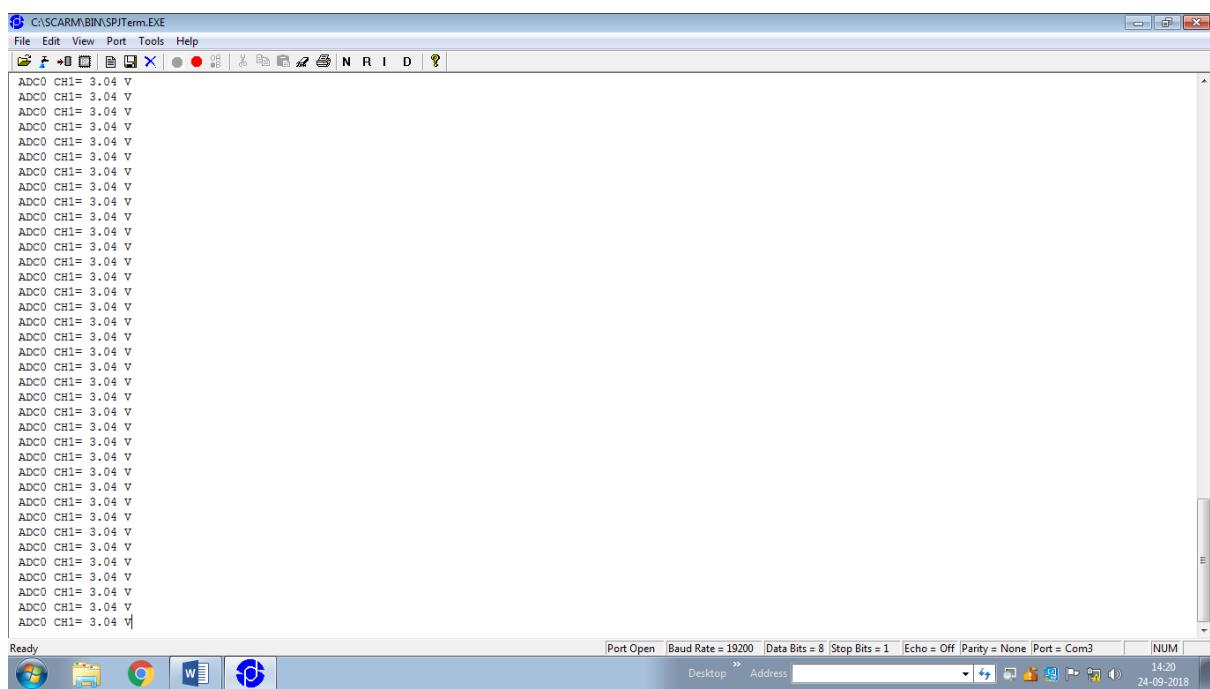
42. The SPJTerminal window will be opened. (Check for the settings given at the right bottom of the window – make sure it matches the same settings as in Flash magic – if there is any changes needed to be done, open the settings of SPJTerminal from the Port option listed on top of that window)



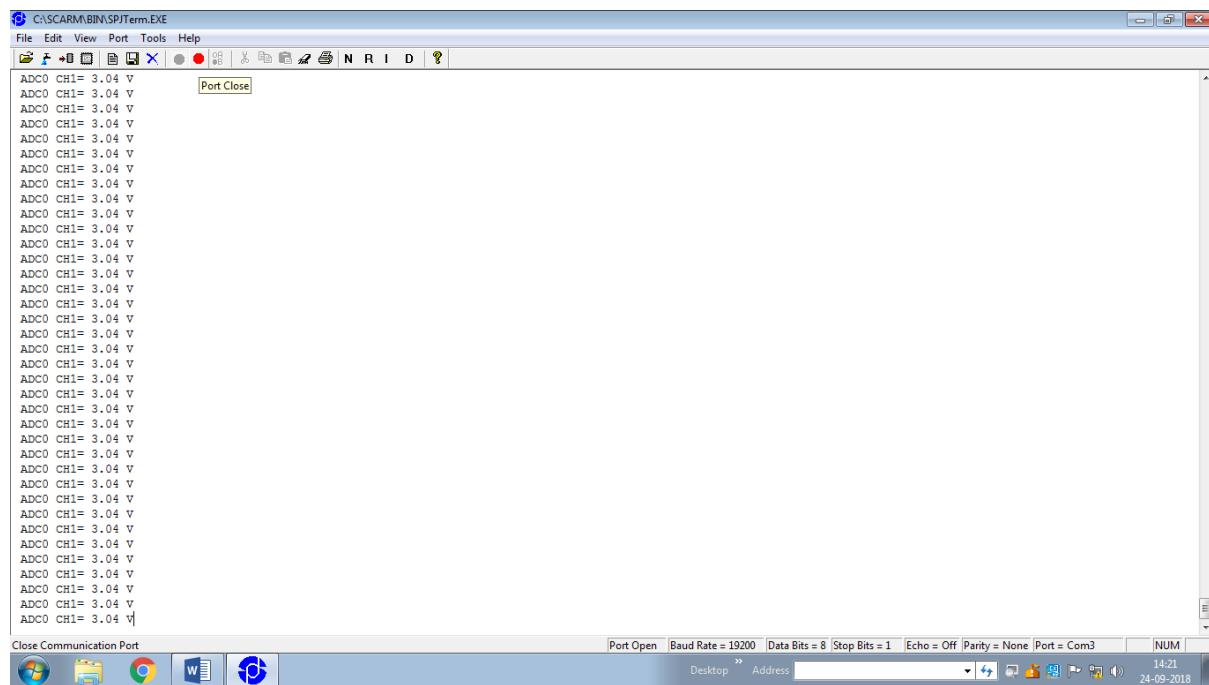
43. Click on green button to open the port.



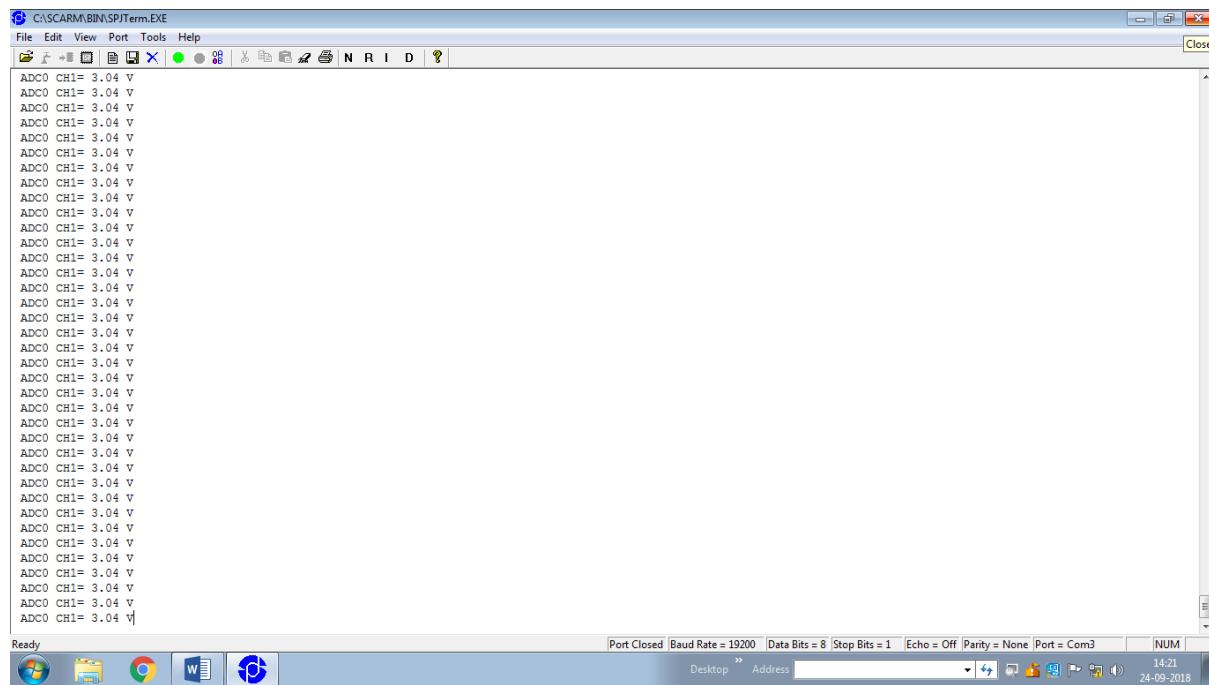
44. The output will be displayed in the window.



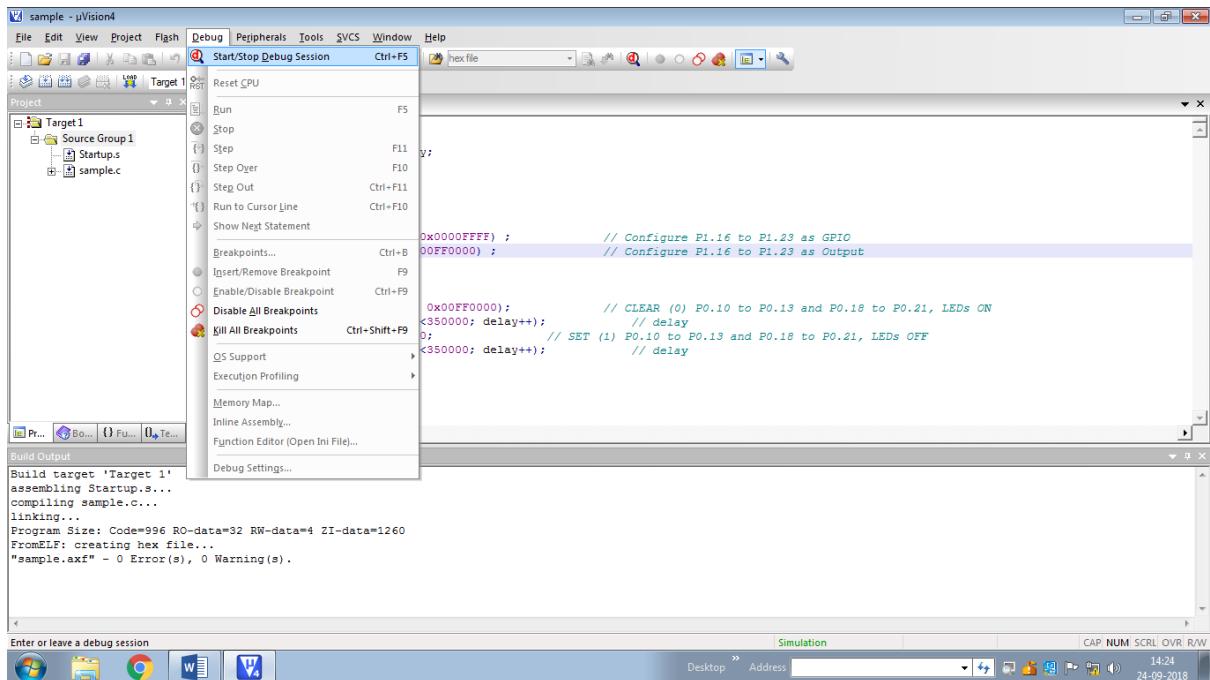
44. To close the port click on red button.



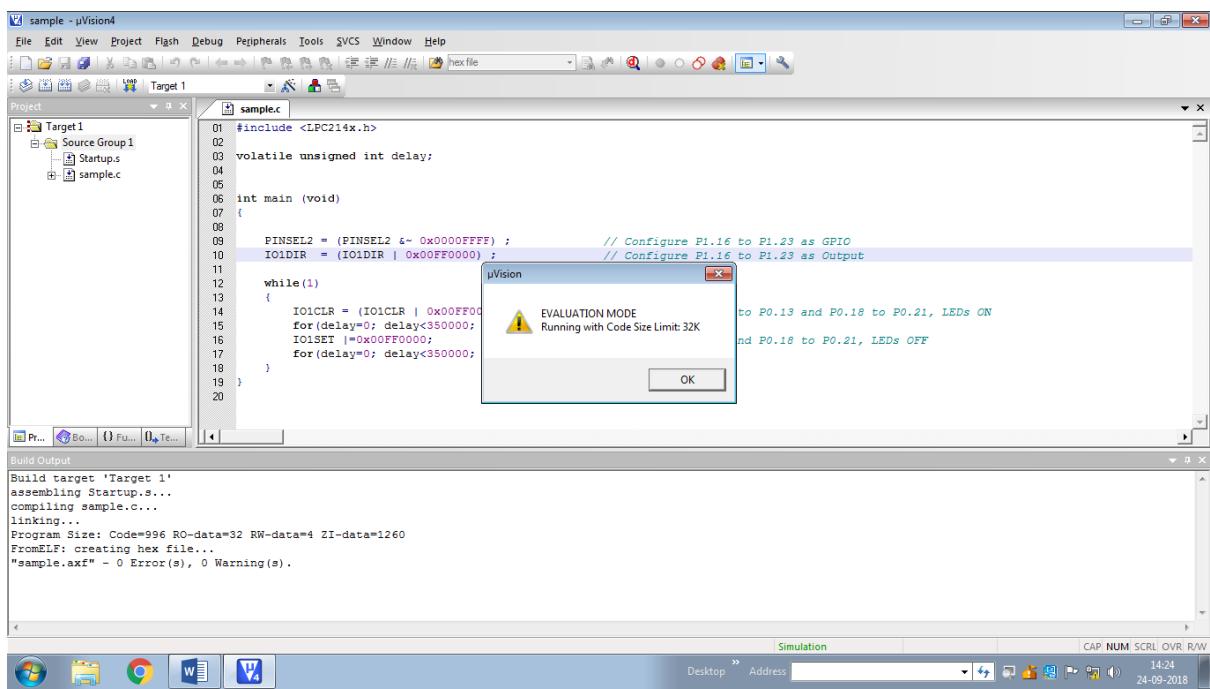
45. Close the SPJTerminal window. (Never close the SPJTerminal window when the port is open)



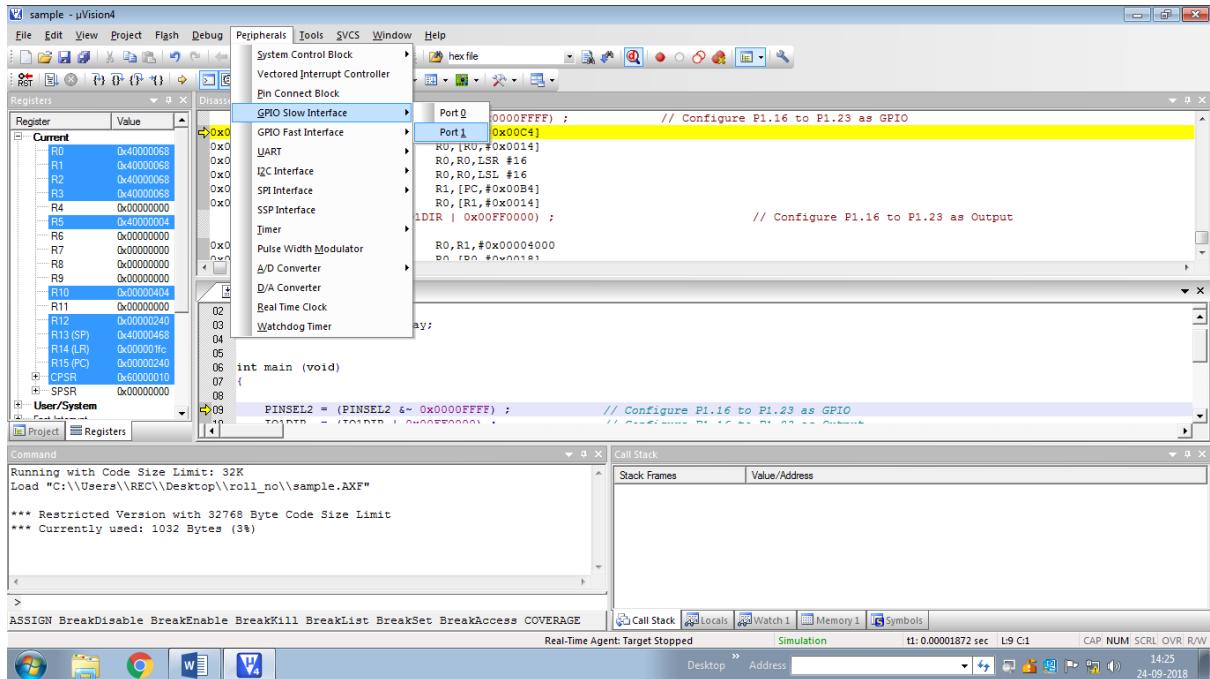
46. For simulation, Go to Debug and select Start/Stop Debug Session (in Keil).



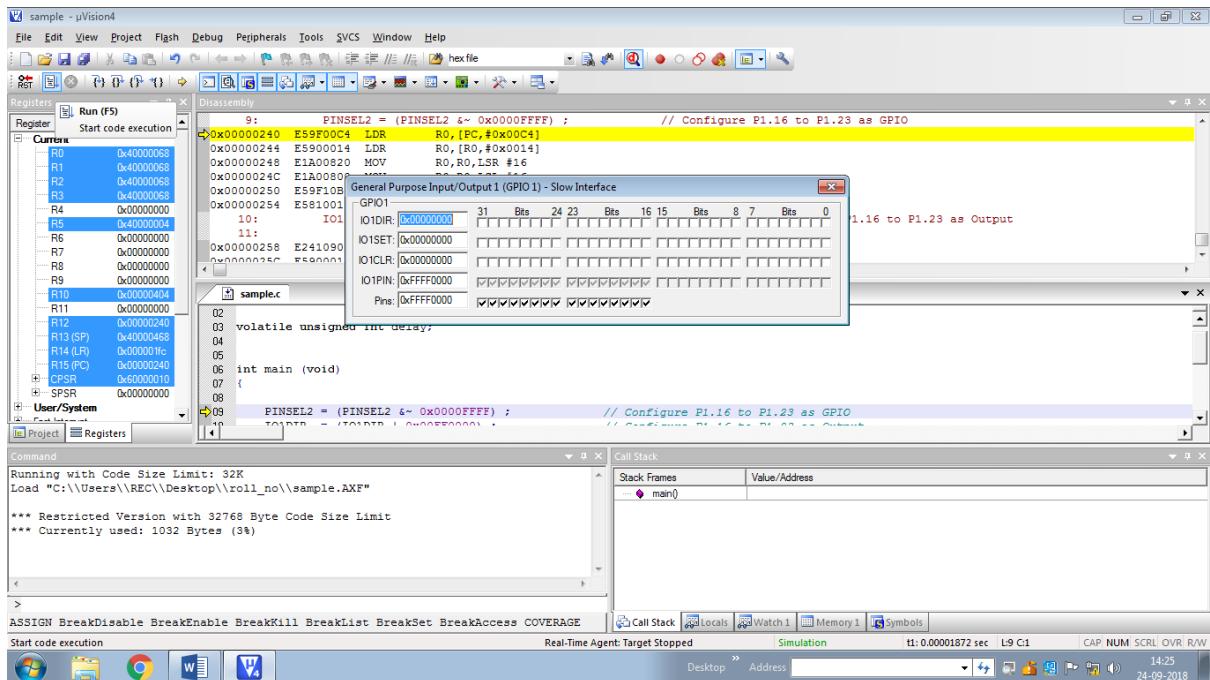
47. Click on OK.



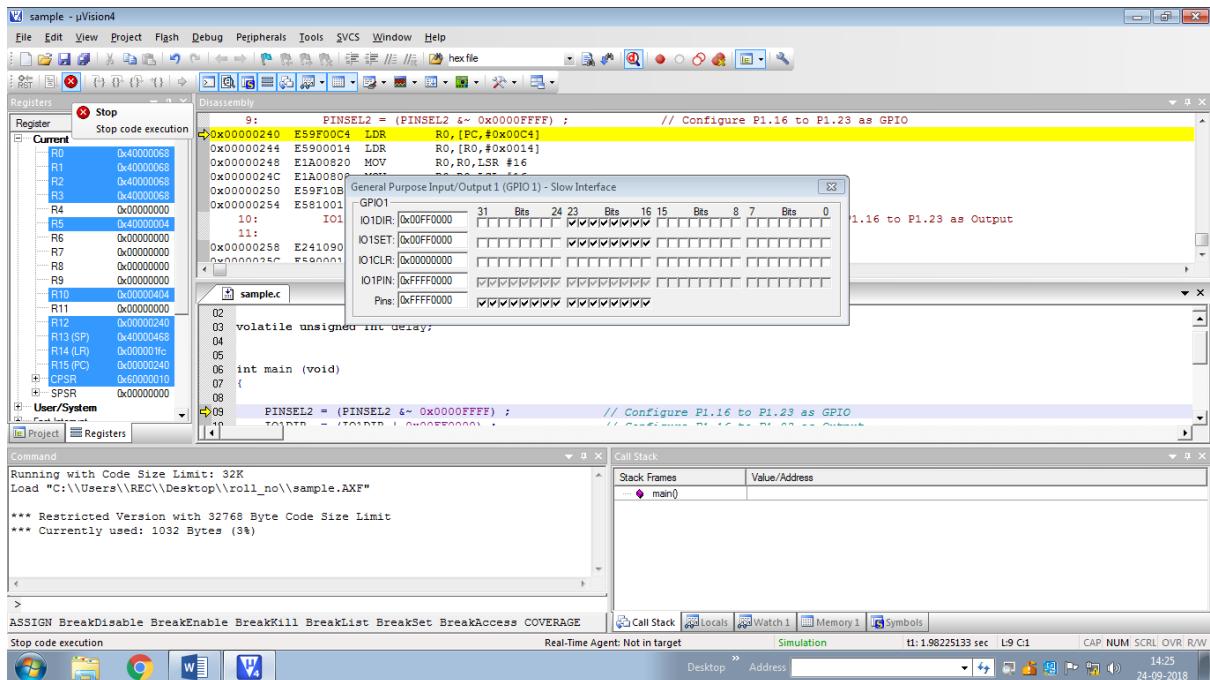
48. Click on peripherals and select GPIO Slow Interface and select the respective port number.



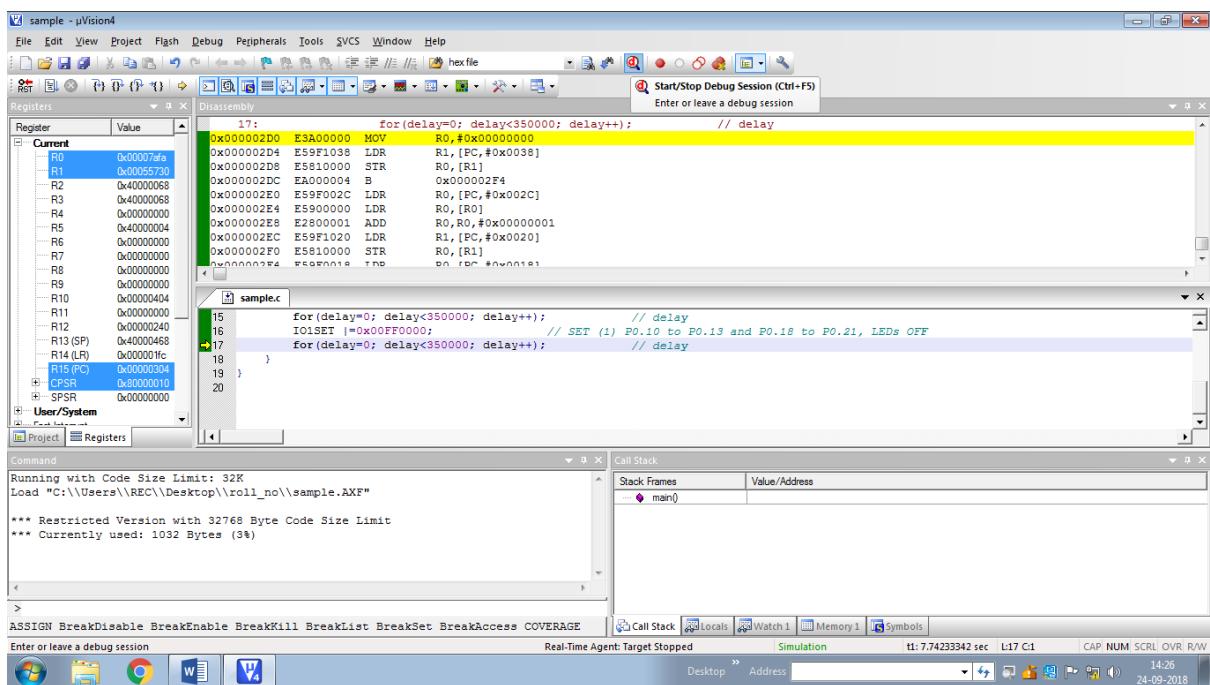
49. Click on the run button, simulated will begin.



50. Click on the stop button to terminate the simulation.



51. To close the debug session click on Start/Stop Debug Session.



## 52. Close the Window.

