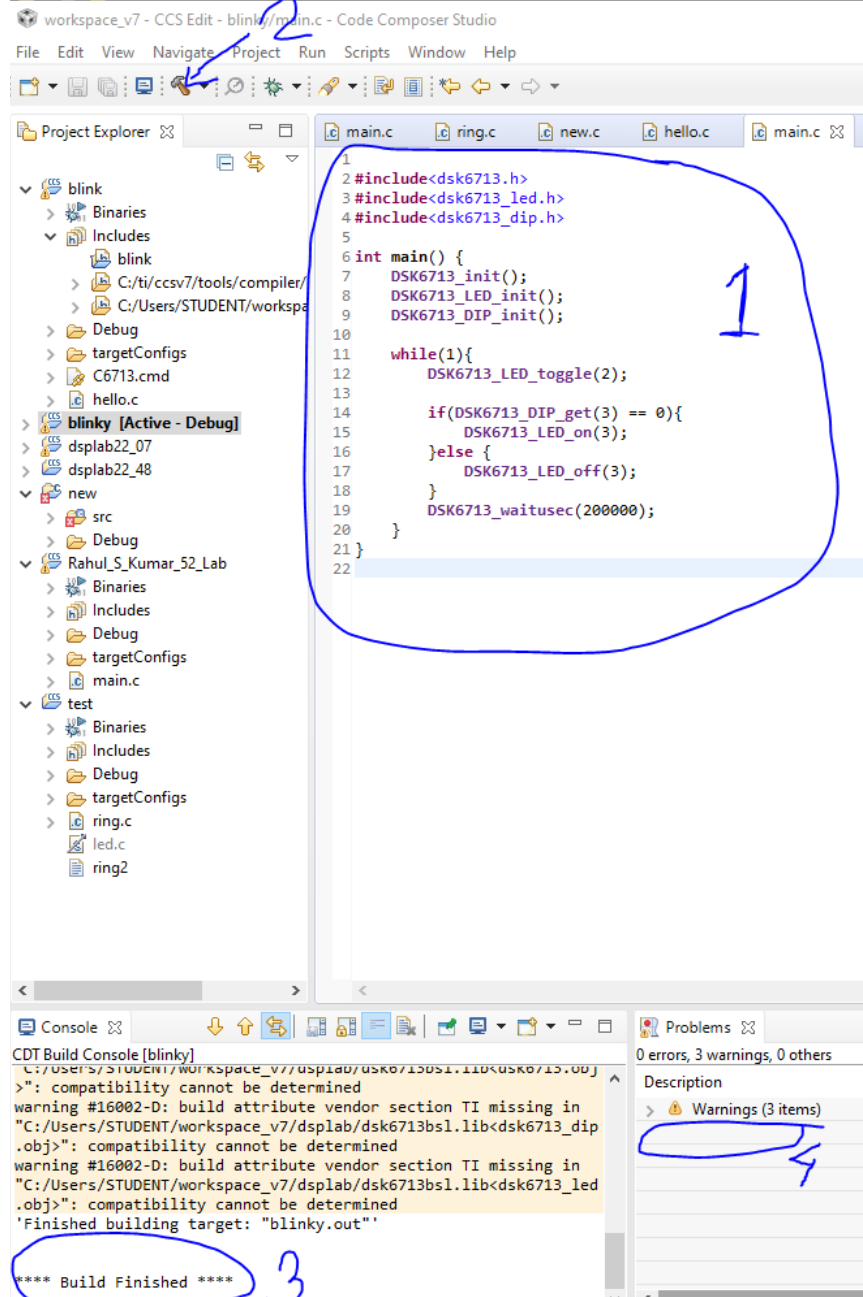


Build and Debug - Code Composer Studio

12 August 2024 12:33

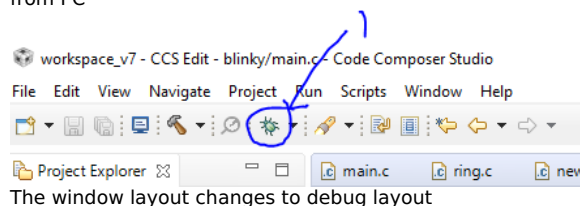
Build a sample application

Paste the code. Build and ensure 0 errors.



Debugging

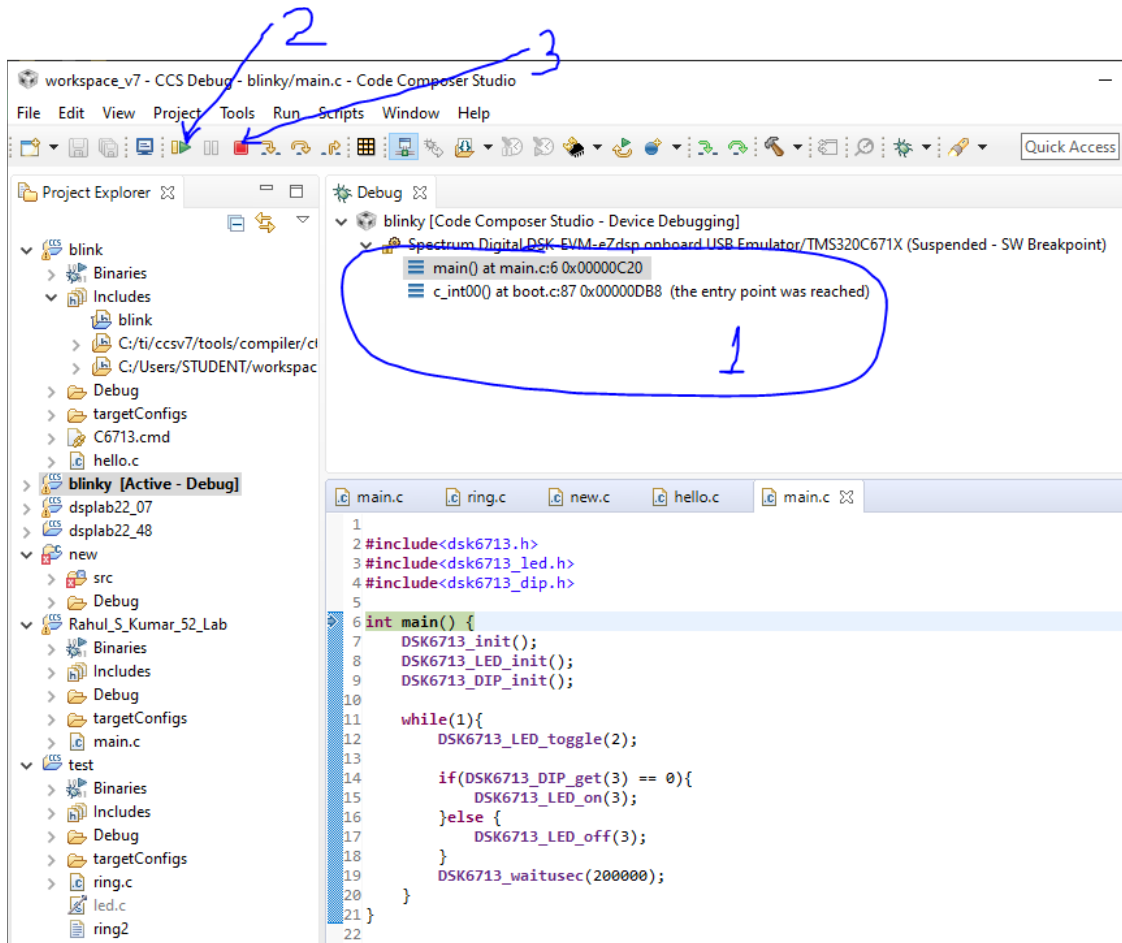
Press debug to load the binary to board using the onboard usb connector connection from PC



The window layout changes to debug layout

The debug layout

1. The current call stack is visible.
2. Press **RUN/RESUME** to start execution/debug. Observe output from board.
3. Press **STOP** to stop execution/debug



Observing Waveforms stored in arrays

Step1

Before starting, note down the variable names to be observed, the type, the size of array, and the point in code where the array contents become valid.

Type	name	size
double	sinWave	[100];
double	squareWave	[1000];
double	convolutionResult	[2 * 1000 - 1];


```

int main() {
    double frequency_sinc = 500.0; // frequency of the sinc wave in Hz
    double frequency_square = 10.0; // frequency of the square wave in Hz
    int sampleRate = 4410; // sample rate in Hz

    generateSinWave(frequency_sinc, 100, sampleRate, sinWave);
    generateSquareWave(frequency_square, 1000, sampleRate, squareWave);
    linearConvolution(sinWave, squareWave, convolutionResult, 1000, 100);
}
  
```

line where variable is updated

Step2

- Enter debug mode.
- Set **breakpoint** after the line where the variable gets updated.
- And run till that line by pressing the **RESUME** button.

Arrow always show the line about to be executed

```

43 int main() {
44     double frequency_sinc = 500.0; // frequency of the sinc wave in Hz
45     double frequency_square = 10.0; // frequency of the square wave in Hz
46
47     int sampleRate = 4410; // sample rate in Hz
48
49
50     generateSinWave(frequency_sinc, 100, sampleRate, sinWave);
51     generateSquareWave(frequency_square, 1000, sampleRate, squareWave);
52
53 }
  
```

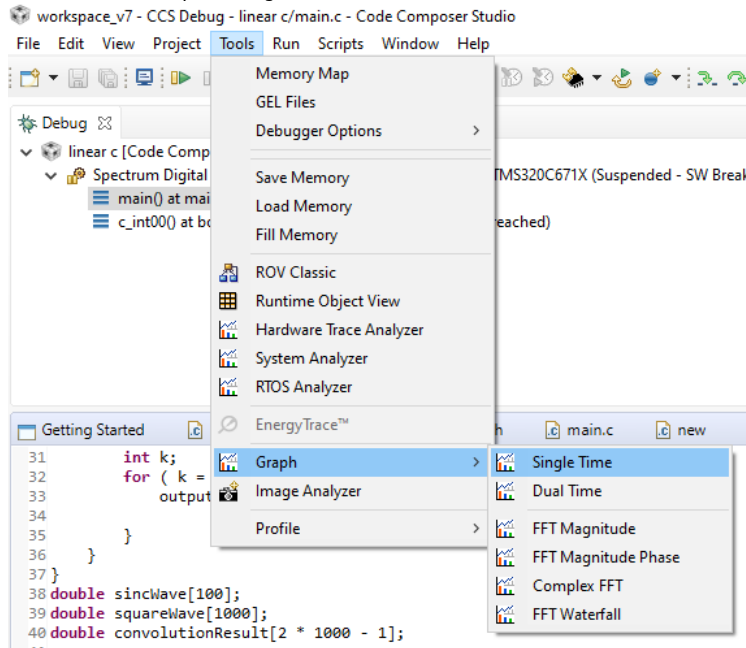
Breakpoint can be setup by double clicking the line number after the line where the variable gets updated. In this case, variable gets updated in line 50, so double click on 51 to set breakpoint.

- The debugger should stop within a second or two at the line of breakpoint (The arrow will shift to line 51).
 - If the debugger doesn't stop within reasonable time (eg. 10s) then press break button and observe the call stack for clue about the problem.
 - Rectify the bug, build and repeat all the steps above.

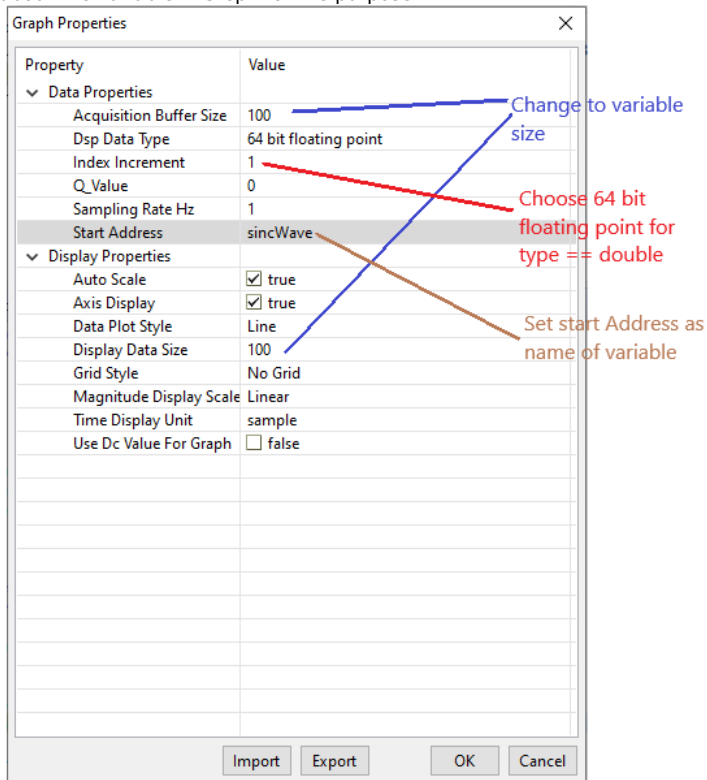
Step3

Once **breakpoint** is hit,

Select Tools > Graph > Single Time



Configure the popup as shown below. We will use the values we already noted down about the variable in Step1 for the purpose.



Expected waveform will be shown in a popup window or tab



