

Debug

Last updated 10/29/20

Debug

- These slides introduce the debugger
- Upon completion: You should be able to use the debugger to run and debug your code

Debug

- Cheap debugger
 - Print out intermediate information
 - `printf("I reached this point");`
 - `printf("foo = %i\n", foo);`
 - Break problems into pieces
 - `foo = a | b << c * d++ - 3 / b % 6;`
 -
 - `foo = d++;`
 - `printf("foo = %i\n", foo);`
 - `foo = c * d++ ;`
 - `printf("foo = %i\n", foo);`
 - ...

Debug

- Example program

```
/*
 * debug_example.c
 *
 * Created on: Dec 17, 2020
 * Author: johnsontimoj
 */
//
// Program to demonstrate debugger
//
//
#include "msp432.h"
#include <stdio.h>

float doublef(float val);
void doublei(int* val_ptr);

int main(void){
    setbuf(stdout, NULL); // added to force printing to flush during debug

    int a;
    int b;
    float c;
    char d;
    a = 2;
    c = 2.5;
    d = 's';

    b = 2 * a;

    printf("%C\n", d);

    d = d + 1;

    printf("Enter a new character:");
    scanf("%c", &d);

    c = doublef(c);

    doublei(&b);

    // Hardware setup
    // Note: pin 5 is Port 4 bit 1
    P4->SEL0 &= ~0x02; // Configure pins as an IO
    P4->SEL1 &= ~0x02;
    P4->DIR |= 0x02; // Output
    P4->OUT &= ~0x02; // Default to low

    // Create squarewave (0.5Hz)
    while(1){
        .....delay_cycles(3000000);
        P4->OUT |= 0x02; // high
        .....delay_cycles(3000000);
        P4->OUT &= ~0x02; // low
    } // end while

    return 0;
} // end main
```

```
float doublef(float val){
    float tmp;
    tmp = val * 2;

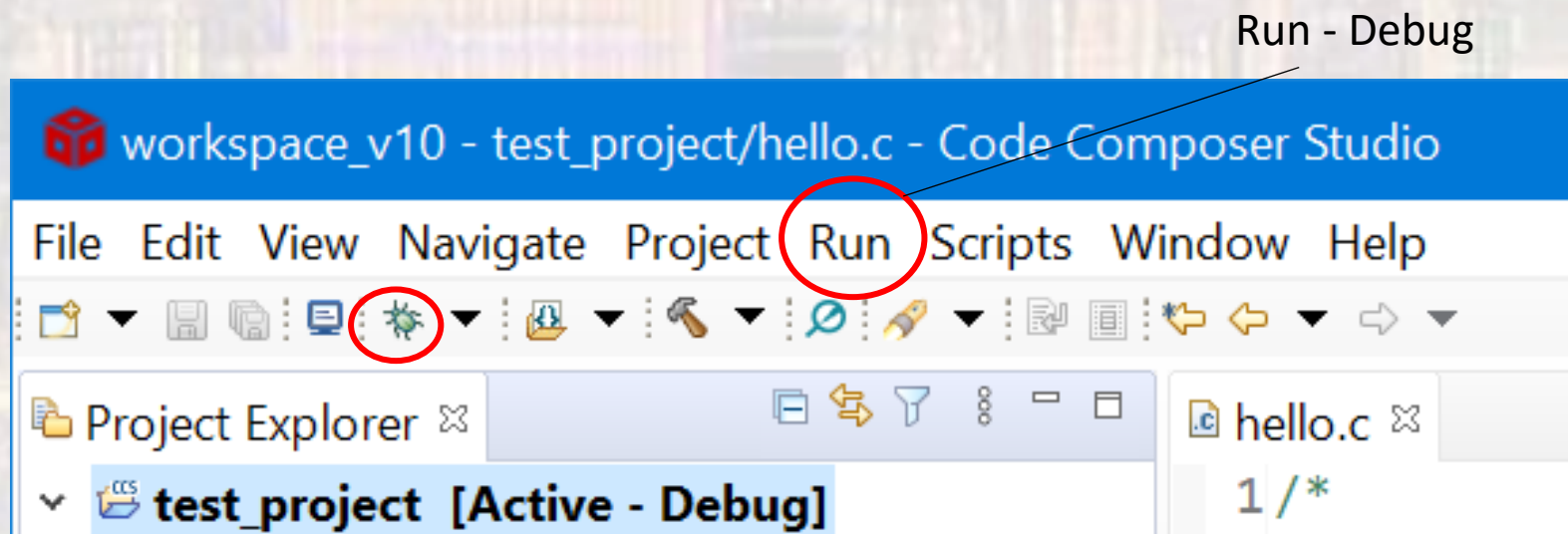
    return tmp;
} // end doublef

void doublei(int* val_ptr){
    int tmp;
    tmp = *val_ptr * 2;
    *val_ptr = tmp;

    return;
} // end doublei
```

Debug

- Debugger
 - Most C tool chains include a debugger
 - Debugger allows
 - Stopping execution
 - Stepping line – by – line
 - Tracking variable values
 - Follow execution into and out of functions



Debug

The screenshot shows the Code Composer Studio interface with the following components:

- Toolbar:** A red circle highlights the Play, Pause, and Stop buttons.
- Debug Console:** Shows the execution state: `Class_MSP_Project [Code Composer Studio - Device Debugging]` and `Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)`. The current execution point is `main() at debug_example.c:18 0x00002B80`.
- Registers Panel:** A table showing variables `a` and `b` with values `0x0` and `0xb` respectively. The values are redacted with "Memory map prevented rea...".
- Source Code:** The file `debug_example.c` is open, showing the `main` function. The cursor is on line 18, which is the line immediately following the `setbuf` call.
- Console:** Shows output from the device: `Class_MSP_Project`, `CORTEX_M4_0: GEL Output: Memory Map Initialization Co...`, `CORTEX_M4_0: GEL Output: Halting Watchdog Timer`, and `CORTEX_M4_0: WARNING : On MSP432P401R hitting a break...`.

Annotations in yellow boxes:

- Play Pause Stop:** Points to the toolbar buttons.
- Program is halted at main:** Points to the current execution point in the debug console.
- Arrow points to the NEXT line to be executed:** Points to the cursor position on line 18 of the source code.

Debug

The screenshot shows the Code Composer Studio interface. The top toolbar includes icons for running, stepping through code, and other debugging actions. The main window is divided into several panes:

- Debug Console:** Shows the current execution state, including the device name (Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0) and the current function being executed (main() at debug_example.c:18).
- Source Code:** Displays the C code for the main function, with line numbers 18 through 28. The code includes variable declarations and assignments.
- Variables, Expressions, and Registers:** A table showing the current state of variables and registers. The table has columns for Name, Type, Value, and Location.

The table in the Variables, Expressions, and Registers window is as follows:

Name	Type	Value	Location
(x)= a	unknown	Memory map prevented rea...	
(x)= b	unknown	Memory map prevented rea...	
(x)= c	unknown	Memory map prevented rea...	
(x)= d	unknown	Memory map prevented rea...	

Annotations in yellow boxes provide further details:

- Variables:** only those in the current scope
- Expressions:** user defined
- Registers:** MSP registers
- currently unknown**
No memory location assigned
- Variables in current scope**

The Console window at the bottom shows the following output:

```
Class_MSP_Project
CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
CORTEX_M4_0: GEL Output: Halting Watchdog Timer
CORTEX_M4_0: WARNING : On MSP432P401R hitting a breakpoint cannot be detected by the debugger when the device is in low power mode.
Click the pause button during debug to check if the device is held at the breakpoint.
```

Debug

The screenshot shows the Code Composer Studio interface. The top toolbar has a red circle around the 'Restart' icon (a circular arrow). A yellow callout box with the text 'restart the system' has an arrow pointing to this icon. Another yellow callout box contains the following text:

- Step Into: Step to the next instruction – go into a function
- Step Over: Step over a function – but execute it
- Step Return: Complete the current element and return

The main window displays a C program in the editor:

```
18 int main(void){
19     setbuf(stdout, NULL); // added to force printing to flush during debug
20
21     int a;
22     int b;
23     float c;
24     char d;
25     a = 2;
26     c = 2.5;
27     d = 's';
28 }
```

The console window at the bottom shows the following output:

```
Class_MSP_Project
CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
CORTEX_M4_0: GEL Output: Halting Watchdog Timer
CORTEX_M4_0: WARNING : On MSP432P401R hitting a breakpoint cannot be detected by the debugger when the device is in low power mode.
Click the pause button during debug to check if the device is held at the breakpoint.
```

Name	Type	Value	Location
(x)=		Memory map prevented rea...	
(x)=		Memory map prevented rea...	
(x)=		Memory map prevented rea...	
(x)=		Memory map prevented rea...	

Debug

Step into or step over

The screenshot shows the Code Composer Studio interface with a debug session. The main window displays the source code for `debug_example.c`. The `main` function is highlighted, and the first executable line, `setbuf(stdout, NULL);`, is selected. The `Variables` window on the right shows the state of variables `a`, `b`, `c`, and `d`. The values are `2`, `17257`, `0.0`, and `0 '\x00'` respectively. The memory locations for these variables are `0x2000FFE8`, `0x2000FFEC`, `0x2000FFF0`, and `0x2000FFF4`. The `Console` window at the bottom shows the output of the program, including a warning message.

Name	Type	Value	Location
(*) a	int	2	0x2000FFE8
(*) b	int	17257	0x2000FFEC
(*) c	float	0.0	0x2000FFF0
(*) d	unsigned char	0 '\x00'	0x2000FFF4

Note: the compiler has already allocated all variables a spot in memory

declared variables now have whatever values happened to be in memory

changes are highlighted in yellow

after 1st step – jumps to 1st executable command

These are yellow because they used to be unknown – now they have been assigned a memory location – and hence have a garbage value

Debug

step over – so we don't go into setbuf function

Executes the setbuf function

No change → white

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)
- main() at debug_example.c:25 0x00002BBE
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain f

Name	Type	Value	Location
(x)- a	int	2	0x2000FFE8
(x)- b	int	17257	0x2000FFEC
(x)- c	float	0.0	0x2000FFF0
(x)- d	unsigned char	0 '\x00'	0x2000FFF4

```
9 // Program to demonstrate debugger
10 //
11 ///////////////////////////////////////////////////////////////////
12 #include "msp432.h"
13 #include <stdio.h>
14
15 float doublef(float val);
16 void doublei(int* val_ptr);
17
18 int main(void){
19     setbuf(stdout, NULL); // added to force printing to flush during debug
20
21     int a;
22     int b;
23     float c;
24     char d;
25     a = 2;
26     c = 2.5;
27     d = ' ';
28
29     b = 2 * a;
30
31     printf("%c\n", d);
32
33     d = d + 1;
```

Console

Class_MSP_Project

CORTEX_M4_0: GEL Output: Memory Map
CORTEX_M4_0: GEL Output: Halting Wait
CORTEX_M4_0: WARNING : On MSP432P401K hitting a breakpoint cannot be detected by the debugger when the device is in low power mode.
Click the pause button during debug to check if the device is held at the breakpoint.

292M of 637M LE

Debug

Step into or step over

Executes the a=2 statement

No change, a was already 2

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- main() at debug_example.c:26 0x00002BC2
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

Name	Type	Value	Location
(x)- a	int	2	0x2000FFE8
(x)- b	int	17257	0x2000FFEC
(x)- c	float	0.0	0x2000FFF0
(x)- d	unsigned char	0 '\x00'	0x2000FFF4

```
10 //
11 ////////////////////////////////////////////////////
12 #include "msp432.h"
13 #include <stdio.h>
14
15 float doublef(float val);
16 void doublei(int* val_ptr);
17
18 int main(void){
19     setbuf(stdout, NULL); // added to force printing to flush during debug
20
21     int a;
22     int b;
23     float c;
24     char d;
25     a = 2;
26     c = 2.5;
27     d = 's';
28
29     b = 2 * a;
30
31     printf("a=%d\n", d);
32
33     d = d + 1;
34
```

Console

Class_MSP_Project

CORTEX_M4_0: GEL Output: Memory Ma
CORTEX_M4_0: GEL Output: Halting M
CORTEX_M4_0: WARNING : On MSP432P...
Click the pause button during debug to check if the device is held at the breakpoint.

307M of 637M

Debug

Step into or step over

Executes the `ac = 2.5` statement

change

jumps to next executable command

The screenshot shows the Code Composer Studio interface during a debug session. The main window displays the source code of `debug_example.c` with a breakpoint at line 27 (`d = 's';`). The Variables window shows the current state of variables: `a` (int) is 2, `b` (int) is 17257, `c` (float) is 2.5, and `d` (unsigned char) is `0 '\x00'`. The Console window shows the output of the program, including a warning message: `WARNING : On MSP432P462, the debugger is not supported in the current mode. Click the pause button during debug to check if the device is held at the breakpoint.`

Name	Type	Value	Location
(0)- a	int	2	0x2000FFE8
(0)- b	int	17257	0x2000FFEC
(0)- c	float	2.5	0x2000FFF0
(0)- d	unsigned char	0 '\x00'	0x2000FFF4

```
11 //////////////////////////////////////////////////
12 #include "msp432.h"
13 #include <stdio.h>
14
15 float doublef(float val);
16 void doublei(int* val_ptr);
17
18 int main(void){
19     setbuf(stdout, NULL); // added to force printing to flush during debug
20
21     int a;
22     int b;
23     float c;
24     char d;
25     a = 2;
26     c = 2.5;
27     d = 's';
28
29     b = 2 * a;
30
31     printf("%d\n", d);
32
33     d = d + 1;
34
35     printf("Enter a new character\n");
36 }
```

Debug

Step into or step over

Executes the `d = 's'` statement

Change
Indicates stored value: 115
And character image: 's'

jumps to next executable command

The screenshot displays the Code Composer Studio interface during a debug session. The top menu bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, and Help. The main workspace is divided into several panes:

- Debug Console:** Shows the current session for 'Class_MSP_Project [Code Composer Studio - Device Debugging]'. It lists the loaded program 'main() at debug_example.c:29' and the current execution point at 'boot_cortex_m.c:121'.
- Variables Window:** A table showing the state of variables:

Name	Type	Value	Location
(0)- a	int	2	0x2000FFE8
(0)- b	int	17257	0x2000FFEC
(0)- c	float	2.5	0x2000FFFO
(0)- d	unsigned char	115 's'	0x2000FFF4
- Code Editor:** Shows the source file 'debug_example.c'. The current line of execution is highlighted at line 29: `b = 2 * a;`. The code includes a `scanf` call that has just finished reading the character 's' into variable `d`.
- Console:** Displays the output from the previous step: `Enter a new character: s`.

Debug

Step into or step over

Executes the `b = 2 * a` statement

Change

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- main() at debug_example.c:31 0x00002BD6
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

(x)= Variables Expressions Registers

Name	Type	Value	Location
(0)= a	int	2	0x2000FFE8
(0)= b	int	4	0x2000FFEC
(0)= c	float	2.5	0x2000FFF0
(0)= d	unsigned char	115's	0x2000FFF4

```
15 float doublef(float val);
16 void doublei(int* val_ptr);
17
18 int main(void){
19     setbuf(stdout, NULL); // added to force printing to flush during debug
20
21     int a;
22     int b;
23     float c;
24     char d;
25     a = 2;
26     c = 2.5;
27     d = 's';
28
29     b = 2 * a;
30
31     printf("%c\n", d);
32
33     d = '0' + 1;
34     printf("Enter a new character: ");
35     scanf("%c", &d);
36
37     c = doublef(c);
38
39 }
```

Console

Class_MSP_Project

CORTEX_M4_0: GEL Output: Memory Map

CORTEX_M4_0: GEL Output: Halting Watchdog

CORTEX_M4_0: WARNING : On MSP432P4011, halting a breakpoint cannot be detected by the debugger when the device is in low power mode. Click the pause button during debug to check if the device is held at the breakpoint.

330M of 637M

LE

Debug

step over – we do not want to go into the print function

Executes the printf function

The screenshot shows the Code Composer Studio interface during a debug session. The main window displays the source code of `debug_example.c` with a breakpoint at line 33. The console window shows the output of the program, which is the character 's'. The callouts explain the step over action and the console output.

```
17
18 int main(void){
19     setbuf(stdout, NULL); // added to force printing to flush during debug
20
21     int a;
22     int b;
23     float c;
24     char d;
25     a = 2;
26     c = 2.5;
27     d = 's';
28
29     b = 2 * a;
30
31     printf("%c\n", d);
32
33     d = d + 1;
34
35     printf("Enter a new character: ");
36     scanf("%c", &d);
37
38     c = doublef(c);
39
40     doublei(&b);
41
```

Console output:

```
Class_MSP_Project:CIO
[CORTEX_M4_0] s
```

jumps to next executable command

Shows the print in the console window

Debug

Step into or step over

Executes the `d = d + 1` statement

Change

jumps to next executable command

The screenshot displays the Code Composer Studio interface during a debug session. The top toolbar includes icons for stepping through code. The 'Debug' window shows the execution context: 'Class_MSP_Project [Code Composer Studio - Device Debugging]' and 'Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)'. The 'main()' function is paused at line 35 of 'debug_example.c'. The 'Variables' window on the right lists variables: 'a' (int, 2), 'b' (int, 4), 'c' (float, 2.5), and 'd' (unsigned char, 116 't'). The 'd' variable is highlighted in yellow. The source code window shows the following code:

```
19  setbuf(stdout, NULL); // added to force printing to flush during debug
20
21  int a;
22  int b;
23  float c;
24  char d;
25  a = 2;
26  c = 2.5;
27  d = 's';
28
29  b = 2 * a;
30
31  printf("%c\n", d);
32
33  d = d + 1;
34
35  printf("Enter a new character: ");
36  scanf("%c", &d);
37
38  c = doublef(c);
39
40  doublei(&b);
41
42  // Hardware setup
43  // Note: nin c is port 4 bit
```

The console window at the bottom shows the output: 'Class_MSP_Project:CIO [CORTEX_M4_0] s'. A yellow callout box points to the 'd' variable value, and another yellow callout box points to the 'scanf' statement in the code.

Debug

step over – we do not want to go into the print function

Executes the printf function

The screenshot shows the Code Composer Studio interface with the following components:

- Code Editor:** Contains the source code for `debug_example.c`. The code is as follows:

```
20
21 int a;
22 int b;
23 float c;
24 char d;
25 a = 2;
26 c = 2.5;
27 d = 's';
28
29 b = 2 * a;
30
31 printf("%c\n", d);
32
33 d = d + 1;
34
35 printf("Enter a new character:");
36 scanf("%c", &d);
37
38 c = doublef(c);
39
40 doublei(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 5
44 P4_VCF1A &= ~0x00000001; // Config
```
- Debugger:** Shows the execution state with a breakpoint at line 36. The `Variables` window displays:

Name	Type	Value	Location
(*)= a	int	2	0x2000FFE8
(*)= b	int	4	0x2000FFEC
(*)= c	float	2.5	0x2000FFF0
(*)= d	unsigned char	116 't'	0x2000FFF4
- Console:** Shows the output of the program:

```
Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character:
```

jumps to next executable command

Shows the print in the console window

Debug

step over – we do not want to go into the scanf function

Executes the scanf function

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug Class_MSP_Project [Code Composer Studio - Device Debugging]
Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Running - Waiting for user input)

Name	Type	Value	Location

```
20
21 int a;
22 int b;
23 float c;
24 char d;
25 a = 2;
26 c = 2.5;
27 d = 's';
28
29 b = 2 * a;
30
31 printf("%c\n", d);
32
33 d = d + 1;
34 printf("Enter a new character:");
35 scanf("%c", &d);
36
37 c = doublef(c);
38 doublei(&b);
39
40 // Hardware setup
41 // Note: pin 5 is Port 4 bit
42 // P4_VFLA & ~P4_VP?? // confi
```

Console
Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character:

216M of 651M Class_MSP_Project:CIO CORTEX...r input LE

It has not moved to the next line of code
It is waiting to complete the scanf (you to enter a value)

Debug

Enter the character - k

completes the scanf function

Change

jumps to next executable command

Entered a 'k'

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)
- main() at debug_example.c:38 0x00002BF8
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

Name	Type	Value	Location
(x)- a	int	2	0x2000FFE8
(x)- b	int	4	0x2000FFEC
(x)- c	float	2.5	0x2000FFF0
(x)- d	unsigned char	107 'k'	0x2000FFF4

```
22 int b;
23 float c;
24 char d;
25 a = 2;
26 c = 2.5;
27 d = 's';
28
29 b = 2 * a;
30 printf("%c\n", d);
31
32 d = d + 1;
33
34 printf("Enter a new character: ");
35 scanf("%c", &d);
36
37 c = doublef(c);
38 doublef(&b);
39
40 // Hardware setup
41 // Note: pin 5 is Port 4 bit 1
42 P4->SEL0 &= ~0x02; // Configure pin5 as an IO
43 P4->SEL1 &= ~0x02;
44 P4->DIR1 |= 0x02; // Configure pin5 as an output
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

221M of 651M

Debug

step into – we want to go into the doublef function

Transfers control to the doublef function

New scope – new variables

Values mean nothing – memory has not yet been assigned for these variables

Parameter is passed via a register

jumps to the doublef function

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- doublef(float) at debug_example.c:60 0x00002C64
- main() at debug_example.c:38 0x00002C00
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

Name	Type	Value	Location
(*) tmp	float	5.60519386e-45 (DEN)	0x2000FFEC
(*) val	float	1.40129846e-45 (DEN)	Register R0

```
debug_example.c
51     __delay_cycles(3000000);
52     P4->OUT |= 0x02; // high
53     __delay_cycles(3000000);
54     P4->OUT &= ~0x02; // low
55 } // end while
56
57 return 0;
58 // end main
59
60 float doublef(float val){
61     float tmp;
62     tmp = val * 2;
63
64     return tmp;
65 } // end doublef
66
67 void doublei(int* val_ptr){
68     int tmp;
69     tmp = *val_ptr * 2;
70     *val_ptr = tmp;
71
72     return;
73 } // end doublei
74
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available
Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

429M of 840M

Debug

Step into or step over

Declares and assigns memory locations for variables

One has garbage
One has the parameter

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- doublef(float)() at debug_example.c:62 0x00002C6C
- main() at debug_example.c:38 0x00002C00
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

debug_example.c

```
51 .....delay_cycles(3000000);
52 P4->OUT |= 0x02; // high
53 .....delay_cycles(3000000);
54 P4->OUT &= ~0x02; // low
55 } // end while
56
57 return 0;
58 } // end main
59
60 float doublef(float val){
61     float tmp;
62     tmp = val * 2;
63
64     return tmp;
65 }// end doublef
66
67 void doublei(int* val_ptr){
68     int tmp;
69     tmp = *val_ptr * 2;
70     *val_ptr = tmp;
71
72     return;
73 }// end doublei
74
```

Name	Type	Value	Location
(x)= tmp	float	4.48415509e-44 (DEN)	0x2000FFE4
(x)= val	float	2.5	0x2000FFE0

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

451M of 840M

Updates Available
Updates are available for your software.
Click to review and install updates.
Set up [Reminder options](#)

Debug

Step into or step over

Executes the multiplication

Change

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- doublef(float()) at debug_example.c:64 0x00002C7C
- main() at debug_example.c:38 0x00002C00
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

(x)= Variables Expressions Registers

Name	Type	Value	Location
(x)= tmp	float	5.0	0x2000FFE4
(x)= val	float	2.5	0x2000FFE0

```
debug_example.c
51     __delay_cycles(3000000);
52     P4->OUT |= 0x02; // high
53     __delay_cycles(3000000);
54     P4->OUT &= ~0x02; // low
55 } // end while
56
57 return 0;
58 } // end main
59
60 float doublef(float val){
61     float tmp;
62     tmp = val * 2;
63
64     return tmp;
65 } // end doublef
66
67 void doublei(int* val_ptr){
68     int tmp;
69     tmp = *val_ptr * 2;
70     *val_ptr = tmp;
71
72     return;
73 } // end doublei
74
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available

Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

465M of 840M

Debug

Step into or step over

Nothing happens

jumps to end of the function

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- doublef(float) at debug_example.c:65 0x00002C80
- main() at debug_example.c:38 0x00002C00
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

Name	Type	Value	Location
tmp	float	5.0	0x2000FFE4
val	float	2.5	0x2000FFE0

```
debug_example.c
51     delay_cycles(3000000);
52     P4->OUT |= 0x02; // high
53     delay_cycles(3000000);
54     P4->OUT &= ~0x02; // low
55 } // end while
56
57 return 0;
58 } // end main
59
60 float doublef(float val){
61     float tmp;
62     tmp = val * 2;
63
64     return tmp;
65 } // end doublef
66
67 void doublei(int* val_ptr){
68     int tmp;
69     tmp = *val_ptr * 2;
70     *val_ptr = tmp;
71
72     return;
73 } // end doublei
74
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available

Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

470M of 840M

Debug

Step into or step over

Completes the function

Highlights them all because they are different than the last screen

Note: c has not changed yet

Indicates it has not done the assignment

Updates Available x
Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

The screenshot shows the Code Composer Studio interface during a debug session. The main editor window displays the source code for 'debug_example.c'. Line 38, 'c = doublef(c);', is highlighted. The 'Variables' window shows the following data:

Name	Type	Value	Location
(0)= a	int	2	0x2000FFE8
(0)= b	int	4	0x2000FFEC
(0)= c	float	2.5	0x2000FFF0
(0)= d	unsigned char	107 'k'	0x2000FFF4

The console window shows the following output:

```
Class_MSP_Project:CIO  
[CORTEX_M4_0] s  
Enter a new character: k
```


Debug

Step into or step over

Completes the assignment

Only shows c has changed

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)

main() at debug_example.c:40 0x00002C04

_c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

Name	Type	Value	Location
(*) a	int	2	0x2000FFE8
(*) b	int	4	0x2000FFEC
(*) c	float	5.0	0x2000FFF0
(*) d	unsigned char	107 'k'	0x2000FFF4

```
30
31 printf("%c\n", d);
32
33 d = d + 1;
34
35 printf("Enter a new character: ");
36 scanf("%c", &d);
37
38 c = doublef(c);
39
40 double c;
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 1
44 P4->SEL0 &= ~0x02; // Configure pins as an IO
45 P4->SEL1 &= ~0x02;
46 P4->DIR |= 0x02; // Output
47 P4->OUT &= ~0x02; // Default to low
48
49 // Create squarewave (0.5Hz)
50 while(1){
51     __delay_cycles(300000);
52     P4->OUT |= 0x02; // high
53     __delay_cycles(300000); // low
54     P4->OUT &= ~0x02;
55 }
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Writable Smart Insert 40 : 17 : 656 499M of 840M

Updates Available x
Updates are available for your software.
Click to review and install updates.
Set up [Reminder options](#)

Debug

step into – we want to go into the doublef function

Transfers control to the doublef function

New scope – new variables

Values mean nothing – memory has not yet been assigned for these variables

Parameter is passed via a register

jumps to the doublef function

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug (x)= Variables Expressions Registers

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- doublef(int *) at debug_example.c:67 0x00002C84
- main() at debug_example.c:44 0x00002C0A
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain f

debug_example.c

```
51     delay_cycles(3000000);
52     P4->OUT |= 0x02; // high
53     delay_cycles(3000000);
54     P4->OUT &= ~0x02; // low
55 } // end while
56
57 return 0;
58 } // end main
59
60 float doublef(float val){
61     float tmp;
62     tmp = val * 2;
63
64     return tmp;
65 } // end doublef
66
67 void doublei(int* val_ptr){
68     int tmp;
69     tmp = *val_ptr * 2;
70     *val_ptr = tmp;
71
72     return;
73 } // end doublei
74
```

Name	Type	Value	Location
(x)- tmp	int	4	0x2000FFEC
val_ptr	int *	0x2000FFEC {4}	Register R0

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available
Updates are available for your software. Click to review and install updates.
Set up [Reminder options](#)

Writable Smart Insert 67 : 1 : 1253 515M of 840M

Debug

Step into or step over

Declares and assigns memory locations for variables

Pointer - arrow
value
value pointed to

One has garbage
One has the parameter

jumps to next executable command

The screenshot shows the Code Composer Studio interface during a debug session. The main window displays the source code for `debug_example.c`. The `doublei` function is highlighted, and the execution is paused at line 69: `tmp = *val_ptr * 2;`. The `Variables` window shows the state of variables: `tmp` (int) at 1084227584 and `val_ptr` (int*) at 0x2000FFEC (4). The `Expressions` window shows the value of `*val_ptr` as 0x2000FFE4. The `Registers` window is empty. The `Console` window shows the prompt `Enter a new character: k`. A yellow box highlights the `return;` statement at line 72, with an arrow pointing to the `next executable command` annotation. Another yellow box highlights the `val_ptr` variable in the `Variables` window, with an arrow pointing to the `Pointer - arrow value value pointed to` annotation. A third yellow box highlights the `val_ptr` variable in the `Expressions` window, with an arrow pointing to the `One has garbage One has the parameter` annotation. A green box at the top right contains the text `Step into or step over`. A red box at the top left contains the text `Declares and assigns memory locations for variables`. A small `Updates Available` dialog box is visible in the bottom right corner.

Name	Type	Value	Location
tmp	int	1084227584	0x2000FFE4
val_ptr	int*	0x2000FFEC (4)	0x2000FFE0

```
51     _delay_cycles(3000000);  
52     P4->OUT |= 0x02; // high  
53     _delay_cycles(3000000);  
54     P4->OUT &= ~0x02; // low  
55 } // end while  
56  
57 return 0;  
58 // end main  
59  
60 float doublef(float val){  
61     float tmp;  
62     tmp = val * 2;  
63  
64     return tmp;  
65 } // end doublef  
66  
67 void doublei(int* val_ptr){  
68     int tmp;  
69     tmp = *val_ptr * 2;  
70     *val_ptr = tmp;  
71  
72     return;  
73 } // end doublei  
74
```

Updates Available
Updates are available for your software. Click to review and install updates.
Set up [Reminder options](#)

Debug

Step into or step over

Executes the multiplication

Change

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- doublei(int*) at debug_example.c:70 0x00002C92
- main() at debug_example.c:44 0x00002C0A
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain f

Name	Type	Value	Location
(*) tmp	int	8	0x2000FFE4
> val_ptr	int*	0x2000FFEC (4)	0x2000FFE0

```
debug_example.c
51     delay_cycles(3000000);
52     P4->OUT |= 0x02; // high
53     delay_cycles(3000000);
54     P4->OUT &= ~0x02; // low
55 } // end while
56
57 return 0;
58 } // end main
59
60 float doublef(float val){
61     float tmp;
62     tmp = val * 2;
63
64     return tmp;
65 } // end doublef
66
67 void doublei(int* val_ptr){
68     int tmp;
69     tmp = *val_ptr * 2;
70     *val_ptr = tmp;
71
72     return;
73 } // end doublei
74
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available x
Updates are available for your software. Click to review and install updates.
Set up [Reminder options](#)

Writable Smart Insert 70 : 1 : 1321 523M of 840M

Debug

Step into or step over

Executes the assignment

Change

jumps to end of the function

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- doublei(int *) at debug_example.c:73 0x00002C98
- main() at debug_example.c:44 0x00002C0A
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

Name	Type	Value	Location
(x) tmp	int	8	0x2000FFE4
val_ptr	int *	0x2000FFEC (8)	0x2000FFE0

```
51     delay_cycles(3000000);
52     P4->OUT |= 0x02; // high
53     delay_cycles(3000000);
54     P4->OUT &= ~0x02; // low
55 } // end while
56
57 return 0;
58 // end main
59
60 float doublef(float val){
61     float tmp;
62     tmp = val * 2;
63
64     return tmp;
65 // end doublef
66
67 void doublei(int* val_ptr){
68     int tmp;
69     tmp = *val_ptr * 2;
70     *val_ptr = tmp;
71
72     return;
73 // end doublei
74
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

471M of 871M

Updates Available

Updates are available for your software. Click to review and install updates.
Set up [Reminder options](#)

Debug

Step into or step over

Completes the function – no assignment to do

This has actually changed

Highlights them all because they are different than the last screen

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- main() at debug_example.c:44 0x00002C0A
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

Name	Type	Value	Location
(*) a	int	2	0x2000FFE8
(*) b	int	8	0x2000FFEC
(*) c	float	5.0	0x2000FFF0
(*) d	unsigned char	107 'k'	0x2000FFF4

```
36 scanf("%c", &d);
37
38 c = doublef(c);
39
40 doublei(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 1
44 P4->SEL0 &= ~0x02; // Configure pin5 as an IO
45 P4->SEL1 &= ~0x02;
46 P4->DIR |= 0x02; // Output
47 P4->OUT &= ~0x02; // Default to low
48
49 // Create squarewaves (0.5Hz)
50 while(1){
51     delay_cycles(3000000);
52     P4->OUT |= 0x02; // high
53     delay_cycles(3000000);
54     P4->OUT &= ~0x02; // low
55 } // end while
56
57 return 0;
58 } // end main
59
float doublef(float val){
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available

Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

493M of 871M

Debug

Open the Registers Tab

Scroll down and expand P4

The screenshot shows the Code Composer Studio interface. The Registers window is open and expanded to show the Port 4 registers. The registers listed are P4IV, P4IN, P4OUT, P4DIR, and P4REN. The values are 0x0000, 0xFF, 0x7C, 0x00, and 0x00 respectively. The descriptions are: Port 4 Interrupt Vector Register [Memory Mapped], Port 4 Input [Memory Mapped], Port 4 Output [Memory Mapped], Port 4 Direction [Memory Mapped], and Port 4 Resistor Enable [Memory Mapped].

Name	Value	Description
P4IV	0x0000	Port 4 Interrupt Vector Register [Memory Mapped]
P4IN	0xFF	Port 4 Input [Memory Mapped]
P4OUT	0x7C	Port 4 Output [Memory Mapped]
P4DIR	0x00	Port 4 Direction [Memory Mapped]
P4REN	0x00	Port 4 Resistor Enable [Memory Mapped]

The main editor shows the following code:

```
28 b = 2 * a;
29
30 printf("%c\n", d);
31
32 d = d + 1;
33
34
35 printf("Enter a new character: ");
36 scanf("%c", &d);
37
38 c = doublef(c);
39
40 doublei(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 1
44 P4->DIR |= 0x02; // Output
45 P4->OUT &= ~0x02; // Default to low
46
47 // Create squarewave (0.5Hz)
48 while(1){
49     delay_cycles(300000);
50     P4->OUT |= 0x02; // high
51     delay_cycles(300000);
52     P4->OUT &= ~0x02; // low
53 }
```

The console shows the output: "Enter a new character: k".

Updates Available: Updates are available for your software. Click to review and install updates. Set up [Reminder options](#).

Values in the Port 4 registers
(arbitrary)

Debug

Step into or step over

Execute P4->DIR |= 0x02

Direction register changes (just bit 1)

Input register changes because bit 1 is no longer an input

jumps to next executable command

Updates Available ✕

Updates are available for your software. Click to review and install updates.

Set up [Reminder options](#)

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)

main() at debug_example.c:45 0x00002F48

_c_int00_noargs() at boot_cortex_m.c:121 0x000044E8 (_c_int00_noargs does not contain fi

Name Value Description

Name	Value	Description
P4		
P4IV	0x0000	Port 4 Interrupt Vector Register [Memc
P4IN	0xFD	Port 4 Input [Memory Mapped]
P4OUT	0x7C	Port 4 Output [Memory Mapped]
P4DIR	0x02	Port 4 Direction [Memory Mapped]
P4REN	0x00	Port 4 Resistor Enable [Memory Mapp

```
29 b = 2 * a;
30
31 printf("%c\n", d);
32
33 d = d + 1;
34
35 printf("Enter a new character: ");
36 scanf("%c", &d);
37
38 c = doublef(c);
39
40 doublei(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 1
44 P4->DIR |= 0x02; // Output
45 P4->OUT &= ~0x02; // Default to low
46
47 // Create square wave (0.5Hz)
48 while(1){
49     delay_cycles(300000);
50     P4->OUT |= 0x02; // high
51     delay_cycles(300000);
52     P4->OUT &= ~0x02; // low
53 }
```

Console

Class_MSP_Project:CIO

[CORTEX_M4_0] s

Enter a new character: k

284M of 888M

Debug

Step into or step over

Execute P4->OUT &= ~0x02

No change in output register – bit 1 was already a 0 (arbitrary)

jumps to next executable command
'While' is not executable – it is a construct

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- main() at debug_example.c:49 0x00002F52
- _c_int00_noargs() at boot_cortex_m.c:121 0x000044E8 (_c_int00_noargs does not contain fi

(x)= Variables Expressions Registers

Name	Value	Description
P4		
P4IV	0x0000	Port 4 Interrupt Vector Register [Memc
P4IN	0xFD	Port 4 Input [Memory Mapped]
P4OUT	0x7C	Port 4 Output [Memory Mapped]
P4DIR	0x02	Port 4 Direction [Memory Mapped]
P4REN	0x00	Port 4 Resistor Enable [Memory Mapp

```
33 d = d + 1;
34 printf("Enter a new character: ");
35 scanf("%c", &d);
36
37
38 c = doublef(c);
39 doublei(&b);
40
41 // Hardware setup
42 // Note: pin 5 is Port 4 bit 1
43 P4->DIR |= 0x02; // Output
44 P4->OUT &= ~0x02; // Default to low
45
46 // Create squarewaves (0.5Hz)
47 while(1){
48     delay_cycles(3000000);
49     P4->OUT |= 0x02; // high
50     delay_cycles(3000000);
51     P4->OUT &= ~0x02; // low
52 } // end while
53
54
55 return 0;
56 } // end main
57
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

303M of 888M

Updates Available

Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

Debug

For `__delay_cycles` must do something special
Right click on the next line -> select run to line

The screenshot shows the Code Composer Studio interface with the following components:

- Debug Console:** Shows the current execution state: `Class_MSP_Project [Code Composer Studio - Device Debugging]`, `Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)`, and `main() at debug_example.c:50 0x00002F60`.
- Registers Panel:** Displays the `P4` register set with values: `P4IV: 0x0000`, `P4IN: 0xFD`, `P4OUT: 0x7C`, `P4DIR: 0x02`, and `P4REN: 0x00`.
- Code Editor:** Shows the source code for `debug_example.c`. The current execution point is at line 50: `P4->OUT |= 0x02; // high`. A yellow callout points to a right-click context menu option `Run to Line` on this line.
- Console:** Shows the output: `Class_MSP_Project:CIO [CORTEX_M4_0] s Enter a new character: k`.
- Updates Available:** A notification box in the bottom right corner indicates that updates are available for the software.

jumps to next executable command

Updates Available
Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

Debug

Step into or step over

Execute P4->OUT |= 0x02

output register – bit 1 set to 1
input matches output

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- main() at debug_example.c:51 0x00002F6A
- _c_int00_noargs() at boot_cortex_m.c:121 0x000044E8 (_c_int00_noargs does not contain fi

Variables Expressions Registers

Name	Value	Description
P4		
P4IV	0x0000	Port 4 Interrupt Vector Register [Mem...
P4IN	0xFF	Port 4 Input [Memory Mapped]
P4OUT	0x7E	Port 4 Output [Memory Mapped]
P4DIR	0x02	Port 4 Direction [Memory Mapped]
P4REN	0x00	Port 4 Resistor Enable [Memory Mapp...

```
debug_example.c setbuf.c
35 printf("Enter a new character:");
36 scanf("%c", &d);
37
38 c = doublef(c);
39
40 doublei(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 1
44 P4->DIR |= 0x02; // output
45 P4->OUT &= ~0x02; // Default to low
46
47 // Create squarewave (0.5Hz)
48 while(1){
49     __delay_cycles(3000000);
50     P4->OUT |= 0x02; // high
51     __delay_cycles(3000000);
52     P4->OUT &= ~0x02; // low
53 } // end while
54
55 return 0;
56 // end main
57
58 float doublef(float val){
59     float tmp;
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available

Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

Writable Smart Insert 51 : 1 : 932 365M of 842M

Debug

For `__delay_cycles` must do something special
Right click on the next line -> select run to line

The screenshot shows the Code Composer Studio interface during a debug session. The main window displays the source code for `debug_example.c`. The code includes hardware setup for Port 4 and a square wave generation loop. A yellow callout box points to the line `P4->OUT |= ~0x02;` with the text "jumps to next executable command". A green callout box at the top right explains that for `__delay_cycles`, a special action is needed: right-clicking on the next line and selecting "run to line". The right-hand pane shows the register view for the P4 peripheral, listing registers like P4IV, P4IN, P4OUT, P4DIR, and P4REN with their current values. The console at the bottom shows the prompt "Enter a new character: k".

```
36 scanf("%c", &d);
37
38 c = doublef(c);
39
40 doublef(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 1
44 P4->DIR |= 0x02; // Output
45 P4->OUT &= ~0x02; // Default to low
46
47 // Create squarewave (1.5Hz)
48 while(1){
49     __delay_cycles(3000000);
50     P4->OUT |= 0x02; // high
51     __delay_cycles(3000000);
52     P4->OUT &= ~0x02; // low
53 } // end while
54
55 return 0;
56 // end main
57
58 float doublef(float val){
59     float tmp;
60     tmp = val * 2;
```

Name	Value	Description
P4		
P4IV	0x0000	Port 4 Interrupt Vector Register [Memory Mapped]
P4IN	0xFF	Port 4 Input [Memory Mapped]
P4OUT	0x7E	Port 4 Output [Memory Mapped]
P4DIR	0x02	Port 4 Direction [Memory Mapped]
P4REN	0x00	Port 4 Resistor Enable [Memory Mapped]

Console: Class_MSP_Project:CIO [CORTEX_M4_0] s Enter a new character: k

Updates Available: Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

jumps to next executable command

Debug

Step into or step over

Execute P4->OUT &= ~0x02

output register – bit 1 set to 0
input matches output

jumps to beginning of while(1) – infinite loop

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)

main() at debug_example.c:48 0x00002F82

_c_int00_noargs() at boot_cortex_m.c:121 0x000044E8 (_c_int00_noargs does not contain f

Registers

Name	Value	Description
P4		
P4IV	0x0000	Port 4 Interrupt Vector Register [Mem
P4IN	0xFD	Port 4 Input [Memory Mapped]
P4OUT	0x7C	Port 4 Output [Memory Mapped]
P4DIR	0x02	Port 4 Direction [Memory Mapped]
P4REN	0x00	Port 4 Resistor Enable [Memory Mapp

```
debug_example.c
36 scanf("%c", &d);
37
38 c = doublef(c);
39
40 doublei(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 1
44 P4->DIR |= 0x02; // Output
45 P4->OUT &= ~0x02; // Default to low
46
47 // Create squarewave (0.5Hz)
48 while(1){
49     delay_cycles(300000);
50     P4->OUT |= 0x02; // high
51     delay_cycles(300000);
52     P4->OUT &= ~0x02; // low
53 } // end while
54
55 return 0;
56 // end main
57
58 float doublef(float val){
59     float tmp;
60     tmp = val * 2;
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available

Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

Writable Smart Insert 48 : 1 : 831 408M of 842M

Debug

Step into or step over

Nothing happens

jumps to next executable command

Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Debug

Class_MSP_Project [Code Composer Studio - Device Debugging]

- Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
- main() at debug_example.c:49 0x00002F52
- ._c_int00_noargs() at boot_cortex_m.c:121 0x000044E8 (._c_int00_noargs does not contain fi

Name	Value	Description
P4		
P4IV	0x0000	Port 4 Interrupt Vector Register [Memc
P4IN	0xFD	Port 4 Input [Memory Mapped]
P4OUT	0x7C	Port 4 Output [Memory Mapped]
P4DIR	0x02	Port 4 Direction [Memory Mapped]
P4REN	0x00	Port 4 Resistor Enable [Memory Mapp

```
debug_example.c setbuf.c
36 scanf("%c", &c);
37
38 c = doublef(c);
39
40 doublef(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 1
44 P4->DIR |= 0x02; // Output
45 P4->OUT &= ~0x02; // Default to low
46
47 // Create squarewave (0.5Hz)
48 while(1){
49     delay_cycles(300000);
50     P4->OUT |= 0x02; // high
51     delay_cycles(300000);
52     P4->OUT &= ~0x02; // low
53 } //end while
54
55 return 0;
56 } // end main
57
58 float doublef(float val){
59     float tmp;
60     tmp = val * 2;
```

Console

Class_MSP_Project:CIO
[CORTEX_M4_0] s
Enter a new character: k

Updates Available

Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)

Writable Smart Insert 49: 1: 850 411M of 842M LE

Additional Things we can do in the debugger

Debug

Instead of single stepping
Right click on a line and select run_to

The screenshot shows the Code Composer Studio interface. The 'Variables' window displays the following data:

Name	Type	Value	Location
(0)= a	int	2	0x2000FFE8
(0)= b	int	4	0x2000FFEC
(0)= c	float	2.5	0x2000FFF0
(0)= d	unsigned char	115 's'	0x2000FFF4

The code editor shows the following C code:

```
17
18 int main(void){
19     setbuf(stdout, NULL); // added to force printing to flush during debug
20
21     int a;
22     int b;
23     float c;
24     char d;
25     a = 2;
26     c = 2.5;
27     d = 's';
28
29     b = 2 * a;
30     printf("%c\n",d);
31
32
33     d = d + 1;
34
35     printf("Enter a new character: ");
36     scanf("%c", &d);
37
38     c = doublef(c);
39
40     doublei(&b);
41 }
```

A yellow callout box with a red border contains the text: "Executes all the commands up-to but not including the line selected". A bracket points from this text to the code lines from 18 to 33, and a red circle highlights the right-click icon on line 33.

The Console window shows the output: "Class_MSP_Project:CIO [CORTEX_M4_0] s".

An "Updates Available" notification is present in the bottom right corner.

Debug

Instead of single stepping

Right click on the blue area next to a line and select toggle breakpoint

Then hit run

Right click on the blue area next to a line and select toggle breakpoint – to turn it off again

Executes all the commands up-to but not including the line with the breakpoint

Breakpoint bubble

The screenshot shows the Code Composer Studio interface. The top menu bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, and Help. The Debug console shows the project 'Class_MSP_Project' and the device 'Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)'. The main window displays the source code for 'debug_example.c' with a breakpoint set on line 33. The code includes a main function with variables a, b, c, and d, and a printf statement. The console output shows the program execution and a warning message: 'On MSP432P401R hitting a breakpoint cannot be detected by the debugger when the device is in low power mode. Click the pause button during debug to check if the device is held at the breakpoint.'

Updates Available X

Updates are available for your software. Click to review and install updates.
Set up [Reminder options](#)

Debug

```
/*
 * debug_example_bp.c
 *
 * Created on: Dec 17, 2020
 * Author: johnsontimoi
 */
////////////////////////////////////
//
// Program to demonstrate debugger
//
////////////////////////////////////
#include "msp432.h"
#include <stdio.h>

int main(void){
    setbuf(stdout, NULL); // added to force printing to flush during debug

    int a;
    char d;


    while(1){
        printf("\nEnter a new character: ");
        scanf("%c", &d);

        switch(d){
            case 'a':
                a = 5;
                break;
            case 'c':
                a = 7;
                break;
            case 'd':
                a = 9;
                break;
            default:
                a = 0;
                break;
        }

        printf("a is: %i\n", a);
    }

    return 0;
} // end main
```

Would like to stop
when a c is entered



Debug

Instead of single stepping

Right click on the blue area next to a line and select toggle breakpoint (a = 7)

Then hit run

Right click on the blue area next to a line and select toggle breakpoint – to turn it off again

Executes until the breakpoint is encountered – then stops in debug mode

Breakpoint bubble

Runs through the loop for a and t
Encounters the breakpoint for c

Updates Available
Updates are available for your software. Click to review and install updates.
Set up [Reminder options](#)

The screenshot shows the Code Composer Studio interface. The main window displays a C program named `debug_example_bp.c` with the following code:

```
24 scanf("%c", &d);
25
26 switch(d){
27     case 'a':
28     case 'b':
29         a = 5;
30         break;
31     case 'c':
32         a = 7;
33         break;
34     case 'd':
35         a = 9;
36         break;
37     default:
38         a = 0;
39         break;
40
41 } // end switch
42 printf("a is: %i\n", a);
43
44 // end while
45
46 return 0;
47 // end main
48
```

The console window at the bottom shows the following output:

```
Class_MSP_Project:CIO
Enter a new character: a
a is: 5

Enter a new character: t
a is: 0

Enter a new character: c
|
```

The code editor shows a breakpoint bubble on line 32, and the console shows the program has stopped at line 42.