

Debugging

Last updated 7/6/21

Debugging

- These slides use a different platform
 - The Mbed debugger does not work on our part

Debugging

- 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);`
 - ...

Debugging

- 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 pin5 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;

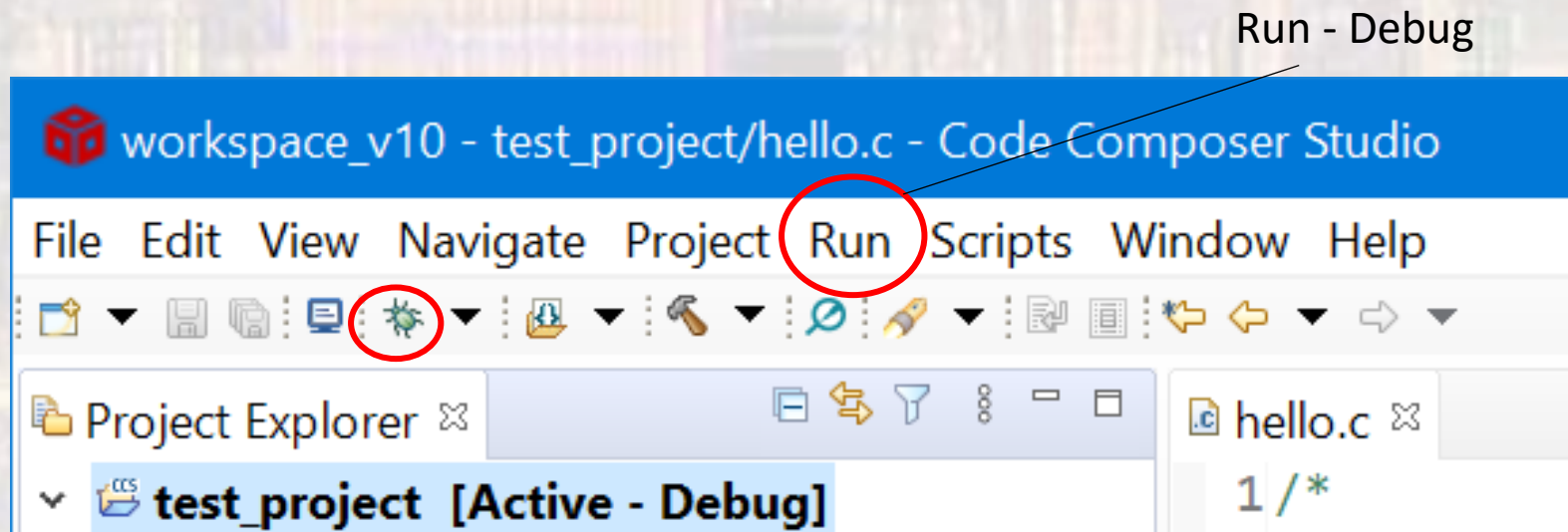
    return tmp;
} // end doublef

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

    return;
} // end doublei
```

Debugging

- 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



Debugging

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

- Toolbar:** The 'Run' button (a green play icon) is circled in red. An arrow points from a yellow callout box to it.
- Callout Box 1:** A yellow rounded rectangle with a red border containing the text "Play Pause Stop".
- Debugger View:** Shows the execution state. The 'main()' function is selected, and the execution point is at line 18 of 'debug_example.c'. An arrow points from a yellow callout box to this line.
- Callout Box 2:** A yellow rounded rectangle with a red border containing the text "Program is halted at main".
- Callout Box 3:** A yellow rounded rectangle with a red border containing the text "Arrow points to the NEXT line to be executed". An arrow points from this box to the line immediately following the current execution point (line 21).
- Registers Panel:** A table showing the state of registers. The 'Value' column contains the text "Memory map prevented rea..." for several entries.
- Code Editor:** Displays the source code for 'debug_example.c'. The current line is highlighted in blue.
- Console:** Shows system output and warnings from the debugger.

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

```
4 * Created on: Dec 17, 2020
5 * Author: johnsantimo1
6 */
7 //////////////////////////////////////////////////
8 //
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 = 's';
28
```

Debugging

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:18 0x00002BB0`.
- Source Code:** Displays the `main` function:

```
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 }
```
- Debugger Windows:** A red circle highlights the `(x)= Variables`, `Expressions`, and `Registers` tabs. Below them is a table of variables:

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...	
- Console:** Shows system messages:

```
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.
```

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**

Debugging

The screenshot shows the Code Composer Studio interface. The top toolbar contains several icons, with the 'Restart System' icon (a circular arrow) circled in red. A yellow callout box with the text 'restart the system' has an arrow pointing to this icon. Below the toolbar, the 'Debug' window shows a tree view of the project and a table of variables. The table has columns for Name, Type, Value, and Location. The 'Value' column contains several entries of 'Memory map prevented rea...'. Below the table, the code editor shows the following C code:

```
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 }
```

At the bottom, the 'Console' window 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.
```

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

Debugging

Step into or step over

The screenshot shows the Code Composer Studio interface. The top menu bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, and Help. The main window is divided into several panes. On the left, the 'Debug' pane shows the project structure: 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:19 0x00002BB6. The 'Variables' pane on the right displays a table of variables:

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

The code editor shows the following code:

```
3 *
4 * Created on: Dec 17, 2020
5 * Author: johnsontimj
6 */
7 ///////////////////////////////////////////////////////////////////
8 //
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 = 'c';
28 }
```

The console pane at the bottom shows the following output:

```
Class_MSP_Project
CORTEX_M4_0: GEL Output: Memory Map
CORTEX_M4_0: GEL Output: Halting Wa
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.
```

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

Debugging

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

Executes the setbuf function

No change → white

jumps to next executable command

The screenshot shows the Code Composer Studio interface during a debugging session. The main window displays the source code of `debug_example.c`. A breakpoint is set at line 25, which is highlighted in green. The console window at the bottom shows the program's output, including a warning about a breakpoint not being detected. The Variables window at the top right shows the current state of variables `a`, `b`, `c`, and `d`.

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 = '\0';
28
29     b = 2 * a;
30
31     printf("%c\n", d);
32
33     d = d + 1;
```

Console Output:

```
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.
```

Debugging

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

```
debug_example.c
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("%s\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

Debugging

Step into or step over

Executes the `ac = 2.5` statement

debug_example.c

```
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 }
```

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

Console

```
Class_MSP_Project
CORTEX_M4_0: GEL Output: Memory Map
CORTEX_M4_0: GEL Output: Halting Wa
CORTEX_M4_0: WARNING : On MSP432P401, setting breakpoints using the debugger is not supported in debug mode.
Click the pause button during debug to check if the device is held at the breakpoint.
```

change

jumps to next executable command

Debugging

Step into or step over

Executes the `d = 's'` statement

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

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:29 0x00002BD0
- _c_int00_noargs() at boot_cortex_m.c:121 0x00004504 (_c_int00_noargs does not contain fi

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

```
debug_example.c
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("%c\n", d);
32
33     d = d + 1;
34
35     printf("Enter a new character: ");
36     scanf("%c", &d);
37 }
```

Console

Class_MSP_Project

CORTEX_M4_0: GEL Output: Memory Map
CORTEX_M4_0: GEL Output: Halting Wait
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.

326M of 637M

Debugging

Step into or step over

Executes the `b = 2 * a` statement

Change

jumps to next executable command

The screenshot shows the Code Composer Studio interface. The main window displays the source code for `debug_example.c`. The code is as follows:

```
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 }
```

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	115's	0x2000FFF4

The Console window shows the following output:

```
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.
```

Debugging

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 a C program being debugged. The program code is as follows:

```
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
```

The Variables window shows the following data:

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

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

Annotations in the image:

- A yellow box points to the `printf` function call, stating "Executes the printf function".
- A yellow box points to the `d = d + 1;` line, stating "jumps to next executable command".
- A yellow box points to the console output, stating "Shows the print in the console window".

Debugging

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 debugging session. The top menu bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, and Help. The main window is divided into several panes:

- Debug Console:** Shows the current execution point at `main() at debug_example.c:35 0x00002BEA`.
- Variables Window:** A table showing the state of variables:

Name	Type	Value	Location
(*)= a	int	2	0x2000FFE8
(*)= b	int	4	0x2000FFEC
(*)= c	float	2.5	0x2000FFF0
(*)= d	unsigned char	116 't'	0x2000FFF4
- Code Editor:** Shows the source code for `debug_example.c`. The current execution point is at line 35, which is highlighted in blue. Line 33, `d = d + 1;`, is highlighted in yellow. The code includes variable declarations for `a`, `b`, `c`, and `d`, and various operations like `b = 2 * a;`, `printf`, and `scanf`.
- Console:** Shows the output of the program, which is the character `s`.

Debugging

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

Executes the printf 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 - HW Breakpoint)

main() at debug_example.c:36 0x00002BF0

_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	2.5	0x2000FFF0
(*)= d	unsigned char	116 't'	0x2000FFF4

```
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
38 c = doublef(c);
39
40 doublei(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit 5
44 P4_VCF1 &= ~0x00000001; // Config
```

Console

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

jumps to next executable command

Shows the print in the console window

Debugging

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
38 c = doublef(c);
39
40 doublei(&b);
41
42 // Hardware setup
43 // Note: pin 5 is Port 4 bit
44 P4_VFLA &= 0x00; // config
```

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)

Debugging

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

```
debug_example.c
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

LE

Debugging

Step into or step over

Declares and assigns memory locations for variables

One has garbage
One has the parameter

jumps to next executable command

The screenshot shows the Code Composer Studio interface. The top menu bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, and Help. The main window is divided into several panes:

- Debug Console:** Shows the execution state of the program. It lists functions like `doublef(float)` at `debug_example.c:62` and `main()` at `debug_example.c:38`.
- Variables Window:** A table showing the current state of variables. It has columns for Name, Type, Value, and Location.

Name	Type	Value	Location
(x)= tmp	float	4.48415509e-44 (DEN)	0x2000FFE4
(x)= val	float	2.5	0x2000FFE0
- Code Editor:** Displays the source code for `debug_example.c`. The current line of execution is highlighted in blue. The code includes a `doublef` function that takes a float parameter and returns a float value.

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

void doublei(int* val_ptr){
    int tmp;
    tmp = *val_ptr * 2;
    *val_ptr = tmp;
    return;
}
```
- Console:** Shows the output of the program, including the prompt `Enter a new character: k`.

Annotations with arrows point from the text boxes to specific elements in the interface:

- The "Step into or step over" box points to the Run button in the top toolbar.
- The "Declares and assigns memory locations for variables" box points to the Variables window.
- The "One has garbage / One has the parameter" box points to the two rows in the Variables window.
- The "jumps to next executable command" box points to the next line of code in the editor.

Updates Available ✕

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

Set up [Reminder options](#)

Debugging

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

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

Debugging

Step into or step over

Nothing happens

The screenshot shows the Code Composer Studio interface. The top menu bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, and Help. The toolbar contains various icons for file operations and debugging. The Debug console shows the current state of the program, including the Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended) and the current execution location at debug_example.c:65. The Variables window displays the current values of variables: tmp (float, 5.0) and val (float, 2.5). The code editor shows the source code for debug_example.c, with the function doublef(float val) highlighted. The Console window shows the output of the program, including the prompt "Enter a new character: k".

```
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 output:

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

jumps to end of the function

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

470M of 840M

LE

Debugging

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

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:38 0x00002C00

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

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

```
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 doublef(&b);
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(3000000);
52     P4->OUT |= 0x02; // high
53     delay_cycles(3000000);
54     P4->OUT &= ~0x02; // low
55 }
```

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](#)

486M of 840M

Debugging

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

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

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

Debugging

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(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(300000);
52     P4->OUT |= 0x02; // high
53     delay_cycles(300000);
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

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

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 LE

Debugging

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

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:69 0x00002C8A
- 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
tmp	int	1084227584	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

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

Writable Smart Insert 69 : 1 : 1296 519M of 840M

Debugging

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 Console

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 fi

(*)= Variables Expressions Registers

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

Debugging

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

(x)= Variables Expressions Registers

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](#)

Debugging

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

The screenshot shows the Code Composer Studio interface. The top menu bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, and Help. The main window is divided into several panes:

- Debug Console:** Shows the execution stack with `main() at debug_example.c:44 0x00002C0A` selected.
- Variables Table:** A table with columns Name, Type, Value, and Location. The values are: `a` (int, 2, 0x2000FFE8), `b` (int, 8, 0x2000FFEC), `c` (float, 5.0, 0x2000FFF0), and `d` (unsigned char, 107 'k', 0x2000FFF4).
- Code Editor:** Shows the source code for `debug_example.c`. The current execution point is at line 44: `P4->SEL0 &= ~0x02; // Configure pin5 as an IO`. The code includes hardware setup for pin 5 and a square wave generation loop.
- Console:** Shows the output of the program: `Class_MSP_Project:CIO [CORTEX_M4_0] s Enter a new character: k`.

Annotations with arrows point to the following elements:

- The **Debug Console** and **Variables Table** are highlighted in yellow, with an arrow pointing to the `main()` entry.
- The **Variables Table** is highlighted in yellow, with an arrow pointing to the `c` and `d` rows.
- The **Code Editor** is highlighted in yellow, with an arrow pointing to line 44.
- The **Console** is highlighted in yellow, with an arrow pointing to the input `k`.

An **Updates Available** dialog box is visible in the bottom right corner, indicating that updates are available for the software.

Debugging

Open the Registers Tab

Scroll down and expand P4

The screenshot shows the Code Composer Studio interface. The Registers window is open, displaying the following data:

Name	Value	Description
P4		
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 source code in the editor shows the following relevant lines:

```
44 P4->DIR |= 0x02; // Output
45 P4->OUT &= ~0x02; // Default to low
48 while(1){
49     delay_cycles(300000);
50     P4->OUT |= 0x02; // high
51     delay_cycles(300000);
52     P4->OUT &= ~0x02; // low
```

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

A yellow callout box with a red border contains the text: "Values in the Port 4 registers (arbitrary)". An arrow points from this box to the register values in the Registers window.

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

Values in the Port 4 registers
(arbitrary)

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

277M of 888M

LE

Debugging

Step into or step over

Execute P4->DIR |= 0x02

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
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
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 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 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

Updates Available

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

284M of 888M

Direction register changes (just bit 1)

Input register changes because bit 1 is no longer an input

jumps to next executable command

Debugging

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

(*)= 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 c = doublef(c);
38 doublei(&b);
39
40 // Hardware setup
41 // Note: pin 5 is Port 4 bit 1
42 P4->DIR |= 0x02; // Output
43 P4->OUT &= ~0x02; // Default to low
44
45 // Create squarewaves (0.5Hz)
46 while(1){
47     delay_cycles(3000000);
48     P4->OUT |= 0x02; // high
49     delay_cycles(3000000);
50     P4->OUT &= ~0x02; // low
51 } // end while
52
53 return 0;
54 } // end main
```

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](#)

Debugging

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. The main window displays the source code for `debug_example.c`. The code includes hardware setup for Port 4 and a loop that toggles the output of Port 4 with a delay of 3000000 cycles. A yellow callout box points to line 50, which is `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 click on the next line -> select run to line". The right-hand side of the interface shows the "Registers" window, which lists various hardware registers for Port 4, such as P4IV, P4IN, P4OUT, P4DIR, and P4REN, along with their current values and descriptions. The console window at the bottom shows the program's output, including the prompt "Enter a new character: k".

```
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
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){
```

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...

Console Output:
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

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

Debugging

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

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

Debugging

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 execution state: Class_MSP_Project [Code Composer Studio - Device Debugging], Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint), and the current location: main() at debug_example.c:52 0x00002F78.
- Registers Panel:** Displays the P4 register set:

Name	Value	Description
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]
- Code Editor:** Shows the source code for `debug_example.c`. Line 52 is highlighted in green, and a yellow callout box points to it with the text "jumps to next executable command". The code includes hardware setup for Port 4 and a square wave generation loop using `__delay_cycles`.

```
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 (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;
```
- Console:** Shows the prompt: Class_MSP_Project:CIO [CORTEX_M4_0] s Enter a new character: k
- Updates Available:** A notification box in the bottom right corner.

jumps to next executable command

Debugging

Step into or step over

Execute P4->OUT &= ~0x02

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;
```

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...

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](#)

output register – bit 1 set to 0
input matches output

jumps to beginning of while(1) – infinite loop

Debugging

Step into or step over

Nothing happens

jumps to next executable command

Name	Value	Description
P4		
P4IV	0x0000	Port 4 Interrupt Vector Register [Memory Mapped]
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 Mapped]

```
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(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;
```

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](#)

Additional Things we can do in the debugger

Debugging

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

The screenshot shows the Code Composer Studio interface. The top menu bar includes File, Edit, View, Project, Tools, Run, Scripts, Window, and Help. The toolbar contains various icons for file operations, execution, and debugging. The main window is divided into several panes:

- 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), main() at debug_example.c:33 0x00002F14, and _c_int00_noargs() at boot_cortex_m.c:121 0x000044E8 (_c_int00_noargs does not contain fi).
- Variables Window:** A table showing the current values of variables:

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
- Code Editor:** Shows the source code for debug_example.c. Line 33, `d = d + 1;`, is highlighted in green. A yellow callout box with a red border points to this line, containing the text: "Executes all the commands up-to but not including the line selected".
- Console:** Shows the output of the program: Class_MSP_Project:CIO [CORTEX_M4_0] s.

An "Updates Available" notification is visible in the bottom right corner, stating: "Updates are available for your software. Click to review and install updates. Set up [Reminder options](#)".

Debugging

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 main window displays a C program named `debug_example.c` with the following 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
31     printf("%c\n", d);
32
33     d = d + 1;
34
35
36
37
38
39
40
41
```

A breakpoint is set on line 33, indicated by a blue bubble. The console output shows the following messages:

```
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.
```

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

Debugging

```
/*
 * 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;

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

    } // end while

    return 0;
} // end main
```

Would like to stop
when a c is entered



Debugging

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 with a switch statement. A breakpoint is set on line 32, which is highlighted in blue. The console window at the bottom shows the program's output: "Enter a new character: a", "a is: 5", "Enter a new character: t", "a is: 0", and "Enter a new character: c". The program has stopped at the breakpoint on line 32.

```
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
```

Console 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
```