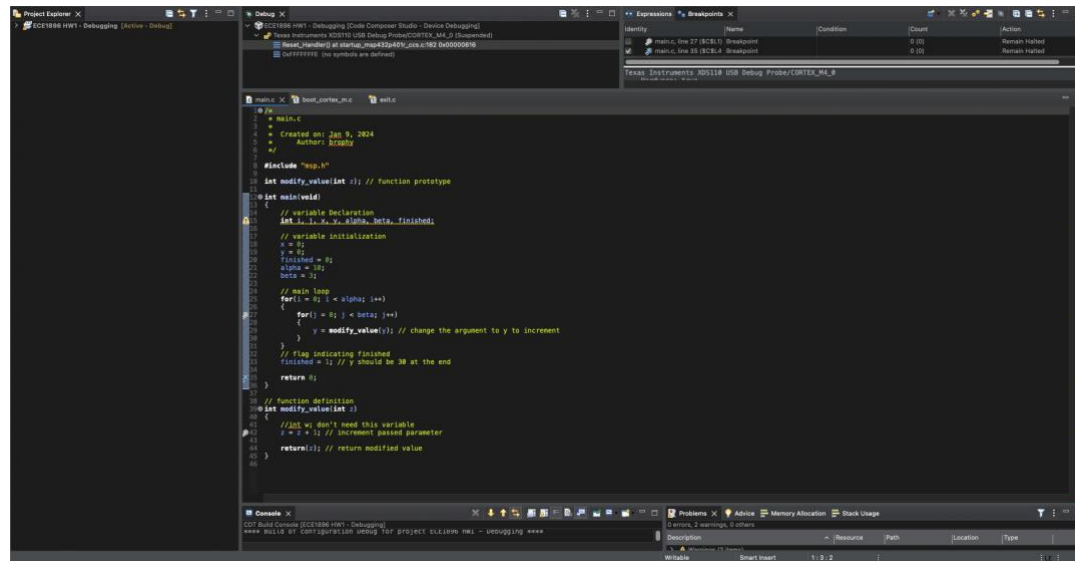


Rick Brophy
ECE1188 Cyberphysical
Dr. Dickerson
Due: 1/16/24

ECE1188 HW1 - Debugging

1.



Zoomed in pic below

```

1  /*
2  * main.c
3  *
4  * Created on: Jan 9, 2024
5  * Author: brophy
6  */
7
8  #include "msp.h"
9
10 int modify_value(int z); // function prototype
11
12 int main(void)
13 {
14     // variable Declaration
15     int i, j, x, y, alpha, beta, finished;
16
17     // variable initialization
18     x = 0;
19     y = 0;
20     finished = 0;
21     alpha = 10;
22     beta = 3;
23
24     // main loop
25     for(i = 0; i < alpha; i++)
26     {
27         for(j = 0; j < beta; j++)
28         {
29             y = modify_value(y); // change the argument to y to increment
30         }
31     }
32     // flag indicating finished
33     finished = 1; // y should be 30 at the end
34
35     return 0;
36 }
37
38 // function definition
39 int modify_value(int z)
40 {
41     //int w; don't need this variable
42     z = z + 1; // increment passed parameter
43
44     return(z); // return modified value
45 }
46

```

Expressions			
Expression	Type	Value	Address
y	int	0x0000001E (Hex)	0x2000FFEC
x	int	0	0x2000FFE8
finished	int	1	0x2000FFF8
i	int	10	0x2000FFE0
j	int	3	0x2000FFE4
alpha	int	10	0x2000FFF0
beta	int	3	0x2000FFF4
+ Add new expression			
Name : y Default:null Hex:0x0000001E Decimal:null Octal:null			

2.

3.

Name	Value	Description
Core Registers		Core Registers
PC	0x0000050E	Program Counter [Core]
SP	0x2000FFE0	General Purpose Register 13 - Stack Pointer [Core]
LR	0x000004ED	General Purpose Register 14 - Link Register [Core]
xPSR	0x21000000	Stores the status of interrupt enables and critical processor status signals [Core]
N	0	Stores bit 31 of the result of the instruction. In other words stores the sign of the number
Z	0	Is set to 1 if the result of the operation is zero else stays 0
C	1	Stores the value of the carry bit if it occurred in an addition or the borrow bit in a subtraction. In a shift stores the last bit shifted out.
V	0	Set to 1 if an overflow occurred
Q	0	Indicates whether an overflow/saturation occurred in the enhanced DSP instructions
ICL_IT_2	00	ICI/IT - bit26-bit25
T	1	Thumb State
RESV	00000000	Reserved
ICL_IT_1	000000	ICI/IT - bit15-bit10
RESV2	0	Reserved
EXCEPTION	00000000	Exception Number
R0	0x00000001	General Purpose Register 0 [Core]
R1	0x0000000A	General Purpose Register 1 [Core]
R2	0x00000004	General Purpose Register 2 [Core]
R3	0x00000000	General Purpose Register 3 [Core]
R4	0x00000000	General Purpose Register 4 [Core]
R5	0x00000000	General Purpose Register 5 [Core]
R6	0x00000000	General Purpose Register 6 [Core]
R7	0x00000000	General Purpose Register 7 [Core]
R8	0x00000000	General Purpose Register 8 [Core]
R9	0x00000000	General Purpose Register 9 [Core]
R10	0x00000000	General Purpose Register 10 [Core]
R11	0x00000000	General Purpose Register 11 [Core]
R12	0x00000638	General Purpose Register 12 [Core]
R13	0x2000FFE0	General Purpose Register 13 [Core]
R14	0x000004ED	General Purpose Register 14 [Core]
MSP	0x2000FFE0	MSP Register [Core]
PSP	0x00000000	PSP Register [Core]
DSP	0x00000000	DSP Register [Core]
CTRL_FAULT_BASE_PRI	0x00000000	CM4 Special Registers [Core]
ADC14		

4.

Memory Browser	
0x2000FFEC	
0x2000FFEC - 0x2000FFEC <Memory Rendering 1>	
32-Bit Hex - T1 Style	
0x2000FFEC	0000000A 00000003 00000001 00005F9
0x2001003C	
0x2001008C	
0x200100DC	
0x2001012C	
0x2001017C	
0x200101CC	
0x2001021C	
0x2001026C	
0x200102BC	
0x2001038C	
0x2001035C	
0x200103AC	
0x200103FC	
0x2001044C	
0x2001049C	
0x200104EC	
0x2001053C	
0x2001058C	
0x200105DC	
0x2001062C	
0x2001067C	
0x200106CC	
0x2001071C	
0x2001076C	
0x200107BC	
0x2001080C	
0x2001085C	
0x200108AC	

Address 0x20010544
Error: Memory map prevented reading 0x20010544

```

main():
000004b4: B500          push    {r14}
000004b6: F1AD0D1C     sub.w   r13, r13, #0x1c
18      x = 0;
000004ba: 2000          movs    r0, #0
000004bc: 9002          str     r0, [r13, #8]
19      y = 0;
000004be: 2000          movs    r0, #0
000004c0: 9003          str     r0, [r13, #0xc]
20      finished = 0;
000004c2: 2000          movs    r0, #0
000004c4: 9006          str     r0, [r13, #0x18]
21      alpha = 10;
000004c6: 200A          movs    r0, #0xa
000004c8: 9004          str     r0, [r13, #0x10]
22      beta = 3;
000004ca: 2003          movs    r0, #3
000004cc: 9005          str     r0, [r13, #0x14]
25      for(i = 0; i < alpha; i++)
000004ce: 2000          movs    r0, #0
000004d0: 9000          str     r0, [r13]
000004d2: 9900          ldr     r1, [r13]
000004d4: 9804          ldr     r0, [r13, #0x10]
000004d6: 4288          cmp     r0, r1
000004d8: DD17          ble     $C$L4
27      for(j = 0; j < beta; j++)
$C$L1:
000004da: 2000          movs    r0, #0
000004dc: 9001          str     r0, [r13, #4]
000004de: 9805          ldr     r0, [r13, #0x14]
000004e0: 9901          ldr     r1, [r13, #4]
000004e2: 4288          cmp     r0, r1
000004e4: DD0A          ble     $C$L3
29      y = modify_value(y); // change the argument to y to increment
$C$L2:
000004e6: 9803          ldr     r0, [r13, #0xc]
000004e8: F000F814     bl      modify_value
000004ec: 9003          str     r0, [r13, #0xc]
27      for(j = 0; j < beta; j++)
000004ee: 9801          ldr     r0, [r13, #4]
000004f0: 1C40          adds    r0, r0, #1
000004f2: 9001          str     r0, [r13, #4]
000004f4: 9805          ldr     r0, [r13, #0x14]
000004f6: 9901          ldr     r1, [r13, #4]
000004f8: 4288          cmp     r0, r1
000004fa: DCF4          bgt     $C$L2
25      for(i = 0; i < alpha; i++)
$C$L3:
000004fc: 9800          ldr     r0, [r13]
000004fe: 1C40          adds    r0, r0, #1
00000500: 9000          str     r0, [r13]
00000502: 9804          ldr     r0, [r13, #0x10]
00000504: 9900          ldr     r1, [r13]
00000506: 4288          cmp     r0, r1
00000508: DCE7          bgt     $C$L1
33      finished = 1; // y should be 30 at the end
$C$L4:
0000050a: 2001          movs    r0, #1
0000050c: 9006          str     r0, [r13, #0x18]
35      return 0;
0000050e: 2000          movs    r0, #0
36  }
00000510: B007          add     r13, #0x1c
00000512: BD00          pop     {pc}
40  {
modify_value():
00000514: F1AD0D08     sub.w   r13, r13, #8

```