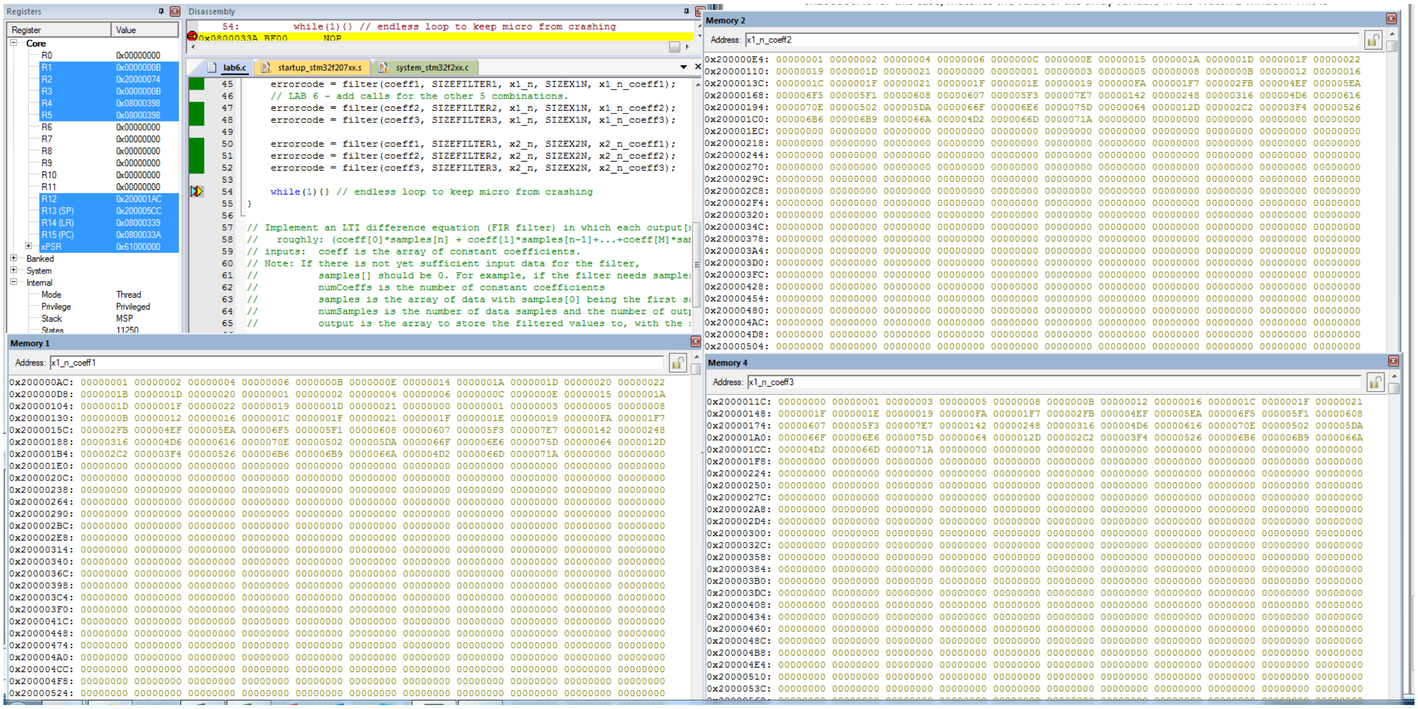
Author: Peter A. Dranishnikov
Lab #: 6
Course: EEL4685C
Due date: March 19th, Spring 2019
Below is the source code, followed by the memory map screenshots.

```
/**
* Author: Peter A. Dranishnikov
* Lab #: 6
* Course: EEL4685C
* Due date: March 19th, Spring 2019
*/
//   Below DOC is for boilerplate template attribution
//  Original Author: David Foster
//   Last modified: 3-14-2018
//  Purpose: Learn to use C in the uVision program for ARM
Cortex series by implementing a simple filter.
//


#include <stm32f2xx.h>


uint32_t coeff1[] = {1, 1, 1, 1};                 // 4-point
simple moving window
uint32_t coeff2[] = {10, 8, 6, 4, 2, 1};     // 6-point weighted
uint32_t coeff3[] = {1, 2, 4, 2, 1};         // 5-point weighted


uint32_t x1_n[]   = {4, 6, 8, 8, 24, 17, 32, 34, 33, 32, 40, 4,
40, 44};                 // first set of data samples
uint32_t x2_n[]       = {1000, 1012, 1040, 2000, 2004, 2080, 0,
2092, 2000, 2003, 1999}; // second set of data samples


// form constants for array sizes
uint32_t SIZEFILTER1 = sizeof(coeff1)/sizeof(uint32_t);
uint32_t SIZEFILTER2 = sizeof(coeff2)/sizeof(uint32_t);
```

```
uint32_t SIZEFILTER3 = sizeof(coeff3)/sizeof(uint32_t);


const uint32_t SIZEX1N = sizeof(x1_n)/sizeof(uint32_t);

const uint32_t SIZEX2N = sizeof(x2_n)/sizeof(uint32_t);




// create space for answers

uint32_t x1_n_coeff1[SIZEX1N];

// LAB 6 - create arrays for the other 5 combinations for
coefficients and input sequences

uint32_t x1_n_coeff2[SIZEX1N];

uint32_t x1_n_coeff3[SIZEX1N];

// ""

uint32_t x2_n_coeff1[SIZEX2N];

uint32_t x2_n_coeff2[SIZEX2N];

uint32_t x2_n_coeff3[SIZEX2N];


// C function prototype - declares the function inputs and
output type so that it may be called below.

uint32_t filter(uint32_t*, uint32_t, uint32_t*, uint32_t,
uint32_t*);




// C programs MUST contain this function, and this is where
execution begins.

int main(void)

{


     volatile uint32_t errorcode;
     errorcode = filter(coeff1, SIZEFILTER1, x1_n, SIZEX1N,
x1_n_coeff1);

     // LAB 6 - add calls for the other 5 combinations.

     errorcode = filter(coeff2, SIZEFILTER2, x1_n, SIZEX1N,
x1_n_coeff2);

     errorcode = filter(coeff3, SIZEFILTER3, x1_n, SIZEX1N,
x1_n_coeff3);


     errorcode = filter(coeff1, SIZEFILTER1, x2_n, SIZEX2N,
x2_n_coeff1);

     errorcode = filter(coeff2, SIZEFILTER2, x2_n, SIZEX2N,
x2_n_coeff2);

     errorcode = filter(coeff3, SIZEFILTER3, x2_n, SIZEX2N,
x2_n_coeff3);


     while(1){} // endless loop to keep micro from crashing

}


// Implement an LTI difference equation (FIR filter) in which
each output[n] is a weighted average of the coeff[i]*samples[n-
i] values

//   roughly: (coeff[0]*samples[n] + coeff[1]*samples[n-
1]+...+coeff[M]*samples[n-M]) / sum(coeff[i]'s)

// inputs:     coeff is the array of constant coefficients.

// Note: If there is not yet sufficient input data for the
filter,

//             samples[] should be 0. For example, if the filter
needs samples[1], samples[0], and samples[-1] for calculating
output[1] for a 3-point filter, then 0 should be used for
samples[-1].

//             numCoeffs is the number of constant coefficients
//              samples is the array of data with samples[0]
being the first sample (oldest)
//              numSamples is the number of data samples and the
number of output values.
//              output is the array to store the filtered values
to, with the same number of values as samples.


uint32_t filter(uint32_t* coeff, uint32_t numCoeffs, uint32_t*
samples, uint32_t numSamples, uint32_t* output)
{
     uint32_t count;
     uint32_t window_sum = 0; //not a hard coded value
     uint32_t coeff_sum = 0;


     //LAB 6 - add code to solve the difference equation passed
to the function. Use the passed lengths and do not hard-code
values.
     for(uint8_t i = 0; i < numCoeffs; i++)
     {
          coeff_sum += coeff[i];
     }
     //The differing weights in a weighted average prevent
significant improvement from a nested loop
     for (count = 0; count < numSamples; count++)
     {
          window_sum = 0;
          for (uint32_t i = numCoeffs; i > 0; i--)
          {
               uint32_t index_sam = count - numCoeffs + i;
               if(index_sam <= count)
               {

                    /**

                    91 92 93 94 95 96 97 98 99 100 101 (sample stream)

                          6  5  4  3  2  1             (window)

                    */


                    window_sum += coeff[numCoeffs - i] * samples[index_sam];

               }

               else

                    break; //terminate inner loop early if window is expended

               /**

                    0 1 2 3 4 5 6 7 (sample stream)

               6 5 4 3 2 1            (window)

               */

          }


          output[count] = window_sum / coeff_sum; //TODO determine if integer division rounding needs to be handled


          //output[count] = (count<<24)+(count<<16)+(count<<8)+count+1;   // 32-bit dummy output to help find array in the memory window.

     }
```

```
    return 0;  // no specific error codes to return yet, so 0 will indicate success (really that errors is false)

}
```



Registers

Register	Value
Core	
R0	0x00000000
R1	0x0000000B
R2	0x20000074
R3	0x0000000B
R4	0x08000398
R5	0x08000398
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x200001AC
R13 (SP)	0x200005CC
R14 (LR)	0x08000339
R15 (PC)	0x0800033A
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	11250

Disassembly

```
    54:         while(1){} // endless loop to keep micro from crashing
0x0800033A BF00        NOP
```

```
    errorcode = filter(coeff1, SIZEFILTER1, x1_n, SIZEX1N, x1_n_coeff1);
    // LAB 6 - add calls for the other 5 combinations.
    errorcode = filter(coeff2, SIZEFILTER2, x1_n, SIZEX1N, x1_n_coeff2);
    errorcode = filter(coeff3, SIZEFILTER3, x1_n, SIZEX1N, x1_n_coeff3);

    errorcode = filter(coeff1, SIZEFILTER1, x2_n, SIZEX2N, x2_n_coeff1);
    errorcode = filter(coeff2, SIZEFILTER2, x2_n, SIZEX2N, x2_n_coeff2);
    errorcode = filter(coeff3, SIZEFILTER3, x2_n, SIZEX2N, x2_n_coeff3);

    while(1){} // endless loop to keep micro from crashing
}

// Implement an LTI difference equation (FIR filter) in which each output[r
//   roughly: (coeff[0]*samples[n] + coeff[1]*samples[n-1]+...+coeff[M]*sar
// inputs:  coeff is the array of constant coefficients.
// Note: If there is not yet sufficient input data for the filter,
//          samples[] should be 0. For example, if the filter needs sample:
//          numCoeffs is the number of constant coefficients
//          samples is the array of data with samples[0] being the first sa
//          numSamples is the number of data samples and the number of outp
//          output is the array to store the filtered values to, with the :
```

Memory 1

Address: x1_n_coeff1

```
0x200000AC: 00000001 00000002 00000004 00000006 0000000B 0000000E 00000014 0000001A 0000001D 00000020 00000022
0x200000D8: 0000001B 0000001D 00000020 00000001 00000002 00000004 00000006 0000000C 0000000E 00000015 0000001A
0x20000104: 0000001D 0000001F 00000022 00000019 0000001D 00000021 00000000 00000001 00000003 00000005 00000008
0x20000130: 0000000B 00000012 00000016 0000001C 0000001F 00000021 0000001F 0000001E 00000019 000000FA 000001F7
0x2000015C: 000002FB 000004EF 000005EA 000006F5 000005F1 00000608 00000607 000005F3 000007E7 00000142 00000248
0x20000188: 00000316 000004D6 00000616 0000070E 00000502 000005DA 0000066F 000006E6 0000075D 00000064 0000012D
0x200001B4: 000002C2 000003F4 00000526 000006B6 000006B9 0000066A 000004D2 0000066D 0000071A 00000000 00000000
0x200001E0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000020C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000238: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000264: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000290: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002BC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002E8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000314: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000340: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000036C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000398: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003C4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003F0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000041C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000448: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000474: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004A0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004CC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004F8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000524: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Memory 2

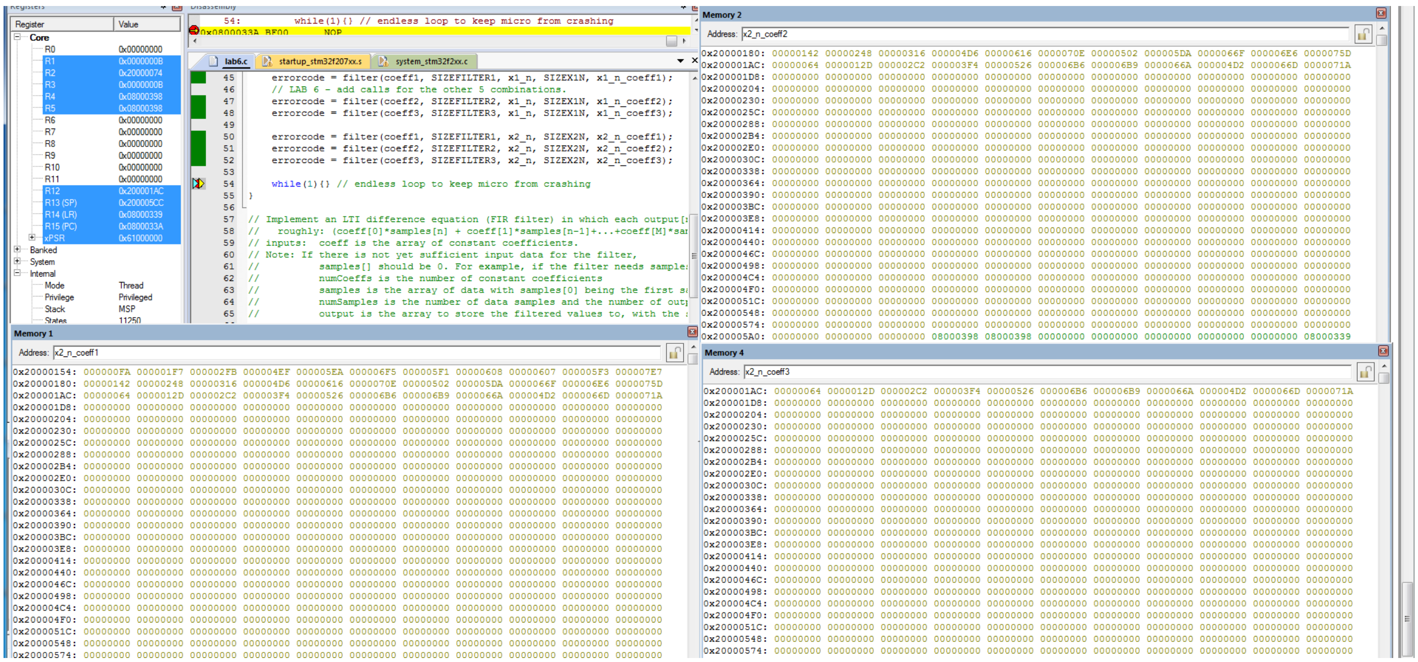
Address: x1_n_coeff2

```
0x200000E4: 00000001 00000002 00000004 00000006 0000000C 0000000E 00000015 0000001A 0000001D 0000001F 00000022
0x20000110: 00000019 0000001D 00000021 00000000 00000001 00000003 00000005 00000008 0000000B 00000012 00000016
0x2000013C: 0000001C 0000001F 00000021 0000001F 0000001E 00000019 000000FA 000001F7 000002FB 000004EF 000005EA
0x20000168: 000006F5 000005F1 00000608 00000607 000005F3 000007E7 00000142 00000248 00000316 000004D6 00000616
0x20000194: 0000070E 00000502 000005DA 0000066F 000006E6 0000075D 00000064 0000012D 000002C2 000003F4 00000526
0x200001C0: 000006B6 000006B9 0000066A 000004D2 0000066D 0000071A 00000000 00000000 00000000 00000000 00000000
0x200001EC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000218: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000244: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000270: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000029C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002C8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002F4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000320: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000034C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000378: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003A4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003D0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003FC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000428: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000454: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000480: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004AC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004D8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000504: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Memory 4

Address: x1_n_coeff3

```
0x2000011C: 00000000 00000001 00000003 00000005 00000008 0000000B 00000012 00000016 0000001C 0000001F 00000021
0x20000148: 0000001F 0000001E 00000019 000000FA 000001F7 000002FB 000004EF 000005EA 000006F5 000005F1 00000608
0x20000174: 00000607 000005F3 000007E7 00000142 00000248 00000316 000004D6 00000616 0000070E 00000502 000005DA
0x200001A0: 0000066F 000006E6 0000075D 00000064 0000012D 000002C2 000003F4 00000526 000006B6 000006B9 0000066A
0x200001CC: 000004D2 0000066D 0000071A 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200001F8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000224: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000250: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000027C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002A8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002D4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000300: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000032C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000358: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000384: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003B0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003DC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000408: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000434: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000460: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000048C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004B8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004E4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000510: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000053C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```



Registers

Register	Value
Core	
R0	0x00000000
R1	0x0000000B
R2	0x20000074
R3	0x0000000B
R4	0x08000398
R5	0x08000398
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x200001AC
R13 (SP)	0x200005CC
R14 (LR)	0x08000339
R15 (PC)	0x0800033A
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	11250

Disassembly

```
    54:         while(1){} // endless loop to keep micro from crashing
0x0800033A BF00        NOP
```

```
    errorcode = filter(coeff1, SIZEFILTER1, x1_n, SIZEX1N, x1_n_coeff1);
    // LAB 6 - add calls for the other 5 combinations.
    errorcode = filter(coeff2, SIZEFILTER2, x1_n, SIZEX1N, x1_n_coeff2);
    errorcode = filter(coeff3, SIZEFILTER3, x1_n, SIZEX1N, x1_n_coeff3);

    errorcode = filter(coeff1, SIZEFILTER1, x2_n, SIZEX2N, x2_n_coeff1);
    errorcode = filter(coeff2, SIZEFILTER2, x2_n, SIZEX2N, x2_n_coeff2);
    errorcode = filter(coeff3, SIZEFILTER3, x2_n, SIZEX2N, x2_n_coeff3);

    while(1){} // endless loop to keep micro from crashing
}

// Implement an LTI difference equation (FIR filter) in which each output[r
//   roughly: (coeff[0]*samples[n] + coeff[1]*samples[n-1]+...+coeff[M]*sar
// inputs:  coeff is the array of constant coefficients.
// Note: If there is not yet sufficient input data for the filter,
//          samples[] should be 0. For example, if the filter needs sample:
//          numCoeffs is the number of constant coefficients
//          samples is the array of data with samples[0] being the first sa
//          numSamples is the number of data samples and the number of outp
//          output is the array to store the filtered values to, with the :
```

Memory 1

Address: x2_n_coeff1

```
0x20000154: 000000FA 000001F7 000002FB 000004EF 000005EA 000006F5 000005F1 00000608 00000607 000005F3 000007E7
0x20000180: 00000142 00000248 00000316 000004D6 00000616 0000070E 00000502 000005DA 0000066F 000006E6 0000075D
0x200001AC: 00000064 0000012D 000002C2 000003F4 00000526 000006B6 000006B9 0000066A 000004D2 0000066D 0000071A
0x200001D8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000204: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000230: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000025C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000288: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002B4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002E0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000030C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000338: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000364: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000390: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003BC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003E8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000414: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000440: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000046C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000498: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004C4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004F0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000051C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000548: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000574: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Memory 2

Address: x2_n_coeff2

```
0x20000180: 00000142 00000248 00000316 000004D6 00000616 0000070E 00000502 000005DA 0000066F 000006E6 0000075D
0x200001AC: 00000064 0000012D 000002C2 000003F4 00000526 000006B6 000006B9 0000066A 000004D2 0000066D 0000071A
0x200001D8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000204: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000230: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000025C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000288: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002B4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002E0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000030C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000338: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000364: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000390: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003BC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003E8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000414: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000440: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000046C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000498: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004C4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004F0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000051C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000548: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000574: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200005A0: 00000000 00000000 00000000 08000398 08000398 00000000 00000000 00000000 00000000 00000000 08000339
```

Memory 4

Address: x2_n_coeff3

```
0x200001AC: 00000064 0000012D 000002C2 000003F4 00000526 000006B6 000006B9 0000066A 000004D2 0000066D 0000071A
0x200001D8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000204: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000230: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000025C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000288: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002B4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200002E0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000030C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000338: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000364: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000390: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003BC: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200003E8: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000414: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000440: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000046C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000498: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004C4: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x200004F0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x2000051C: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000548: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0x20000574: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```