

In the code I measure the ADC values via 3 channels. See below:

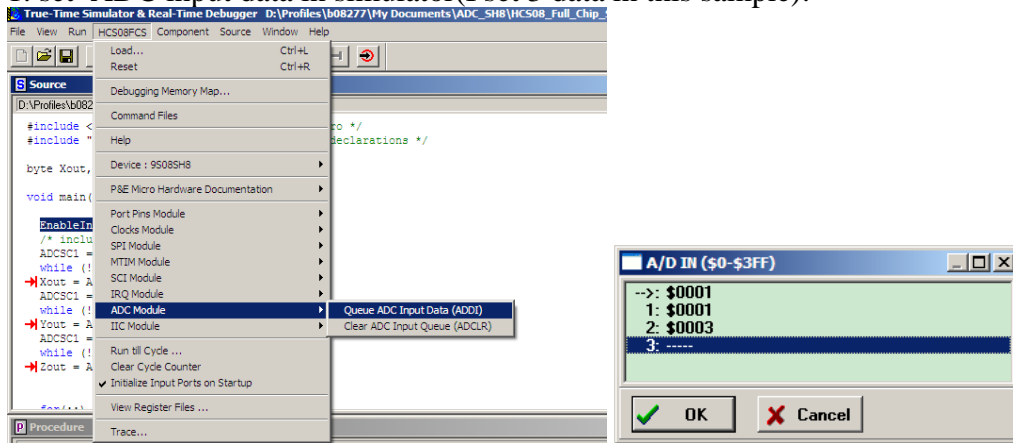
```
byte Xout,Yout,Zout;

void main(void) {

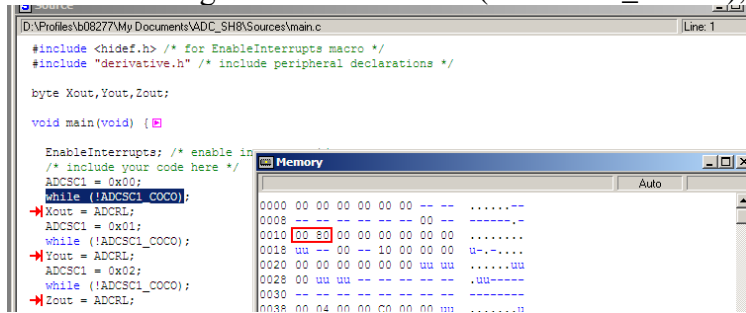
    EnableInterrupts; /* enable interrupts */
    /* include your code here */
    ADCSC1 = 0x00;
    while (!ADCSC1_COCO);
    Xout = ADCRL;
    ADCSC1 = 0x01;
    while (!ADCSC1_COCO);
    Yout = ADCRL;
    ADCSC1 = 0x02;
    while (!ADCSC1_COCO);
    Zout = ADCRL;
}
```

I would list my testing procedure step by step:

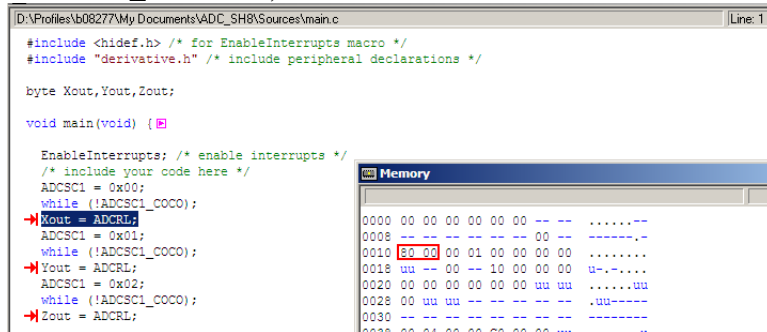
1. set ADC input data in simulator(I set 3 data in this sample):



2. when running code in first “while (!ADCSC1\_COCO);” ADACT=1

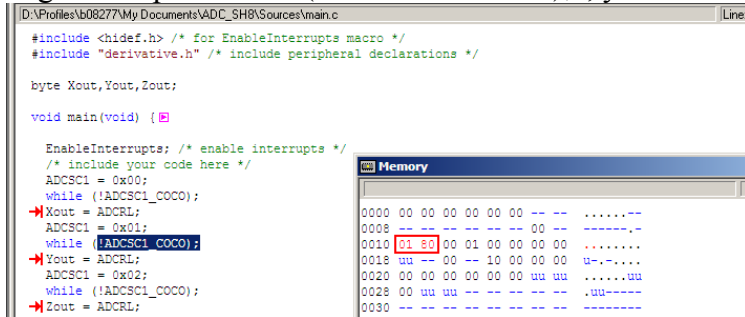


3. Then run to first breakpoint “Xout = ADCRL;” see registers are changed as below: ADCSC1\_COCO=1, ADACT=0



4. then step to `ADCSC1 = 0x01`; you could see `ADCSC1_COCO=0` as the ADC data is read.

5. go on step to “`while (!ADCSC1_COCO);`”, you can see `ADACT=1` again.



The screenshot shows a debugger window with a C source file and a memory window. The source code is as follows:

```
D:\Profiles\h08277\My Documents\ADC_SH8\Sources\main.c |Line:
#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */

byte Xout,Yout,Zout;

void main(void) {
    EnableInterrupts; /* enable interrupts */
    /* include your code here */
    ADCSC1 = 0x00;
    while (!ADCSC1_COCO);
    -> Xout = ADCRL;
    ADCSC1 = 0x01;
    while (!ADCSC1_COCO);
    -> Yout = ADCRL;
    ADCSC1 = 0x02;
    while (!ADCSC1_COCO);
    -> Zout = ADCRL;
```

The memory window shows the following data:

Address	Value
0000	00 00 00 00 00 00 -- -- . . . . .
0008	-- -- -- -- -- 00 -- -- . . . . .
0010	01 00 00 01 00 00 00 00 . . . . .
0018	uu -- 00 -- 10 00 00 00 u- . . . . .
0020	00 00 00 00 00 00 uu uu . . . . .uu
0028	00 uu uu -- -- -- -- .uu-----
0030	-- -- -- -- -- -- -- . . . . .