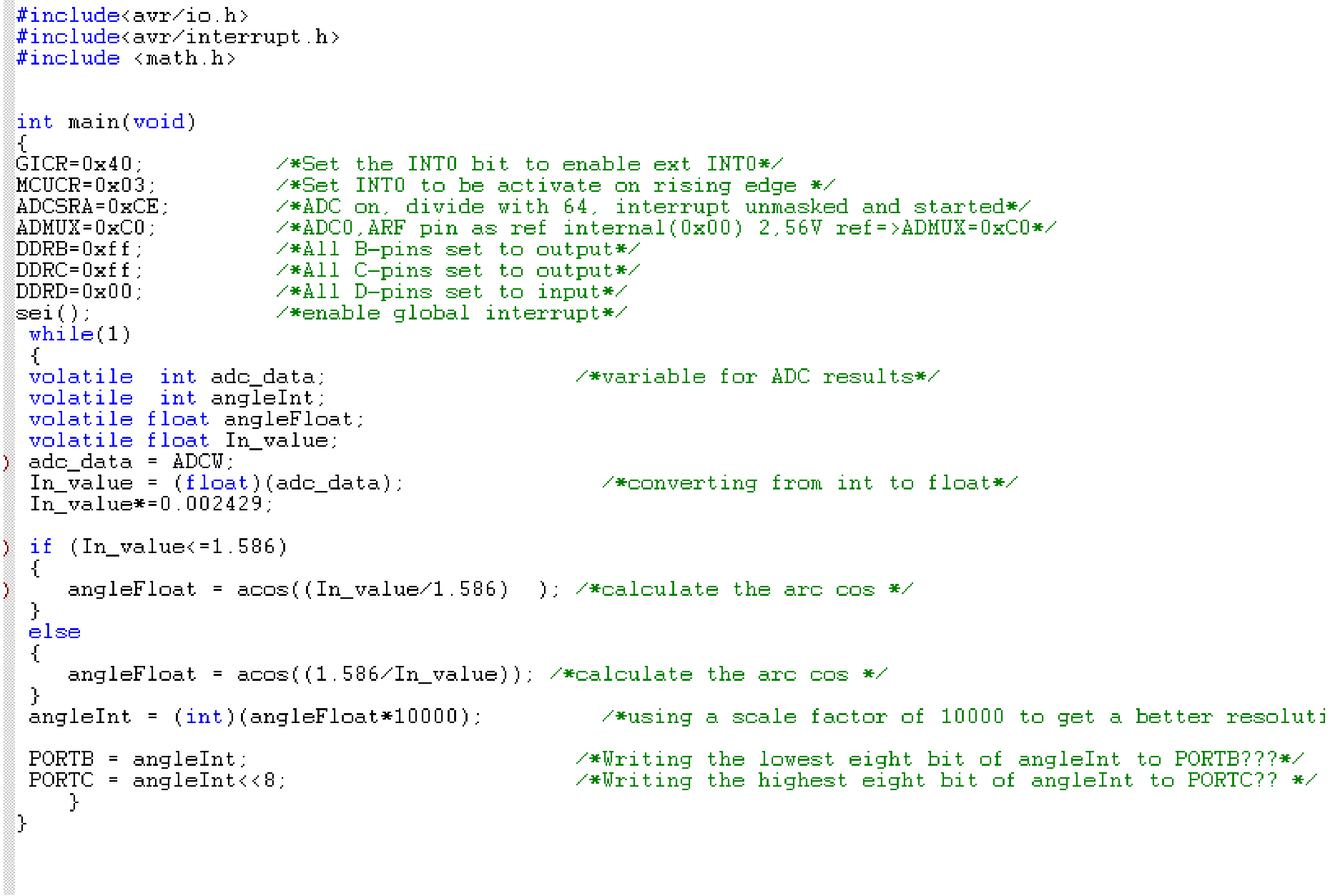
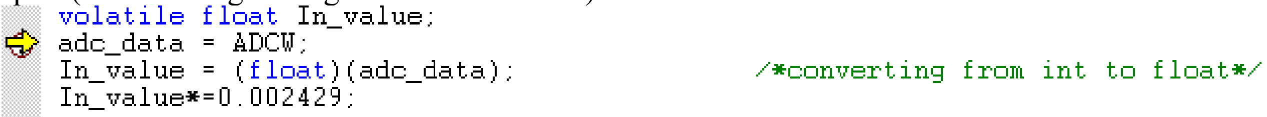
The code is as follows:



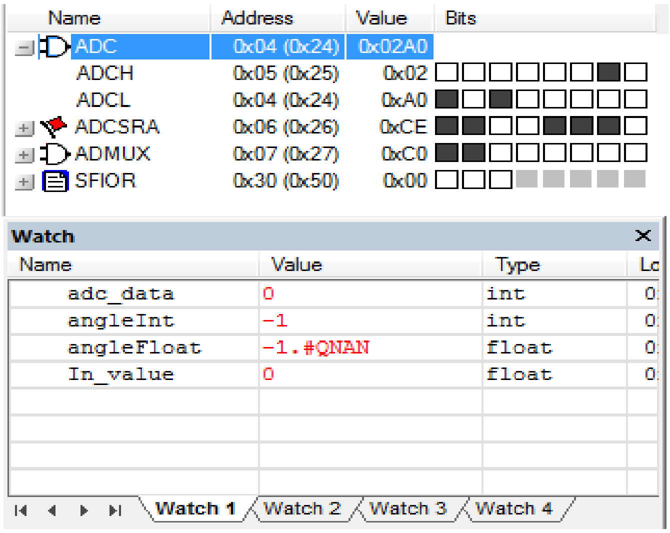
```
#include<avr/io.h>
#include<avr/interrupt.h>
#include <math.h>


int main(void)
{
GICR=0x40;             /*Set the INT0 bit to enable ext INT0*/
MCUCR=0x03;            /*Set INT0 to be activate on rising edge */
ADCSRA=0xCE;           /*ADC on, divide with 64, interrupt unmasked and started*/
ADMUX=0xC0;            /*ADC0,ARF pin as ref internal(0x00) 2,56V ref=>ADMUX=0xC0*/
DDRB=0xff;             /*All B-pins set to output*/
DDRC=0xff;             /*All C-pins set to output*/
DDRD=0x00;             /*All D-pins set to input*/
sei();                 /*enable global interrupt*/
 while(1)
 {
 volatile  int adc_data;                     /*variable for ADC results*/
 volatile  int angleInt;
 volatile float angleFloat;
 volatile float In_value;
 adc_data = ADCW;
 In_value = (float)(adc_data);                 /*converting from int to float*/
 In_value*=0.002429;

 if (In_value<=1.586)
 {
    angleFloat = acos((In_value/1.586)  ); /*calculate the arc cos */
 }
 else
 {
    angleFloat = acos((1.586/In_value)); /*calculate the arc cos */
 }
 angleInt = (int)(angleFloat*10000);          /*using a scale factor of 10000 to get a better resoluti

 PORTB = angleInt;                          /*Writing the lowest eight bit of angleInt to PORTB???*/
 PORTC = angleInt<<8;                       /*Writing the highest eight bit of angleInt to PORTC?? */
    }
}
```

Step 1: (after reading the registers and variables)



```
volatile float In_value;
adc_data = ADCW;
In_value = (float)(adc_data);                 /*converting from int to float*/
In_value*=0.002429;
```

The add watch for step 1: (presetting ADC register to 0x02A0)

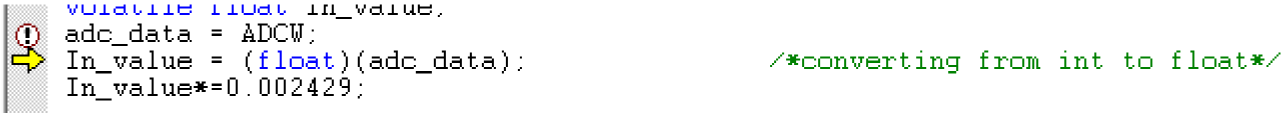


Name	Address	Value	Bits
ADC	0x04 (0x24)	0x02A0	
ADCH	0x05 (0x25)	0x02	□□□□□□■□
ADCL	0x04 (0x24)	0xA0	■□■□□□□□
ADCSRA	0x06 (0x26)	0xCE	■■□□■■■□
ADMUX	0x07 (0x27)	0xC0	■■□□□□□□
SFIOR	0x30 (0x50)	0x00	□□□▨▨▨▨▨

Watch

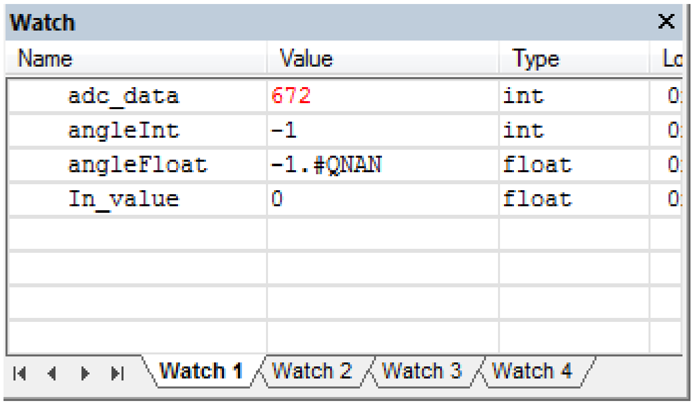
Name	Value	Type	Lo
adc_data	0	int	0:
angleInt	-1	int	0:
angleFloat	-1.#QNAN	float	0:
In_value	0	float	0:

Step 2:



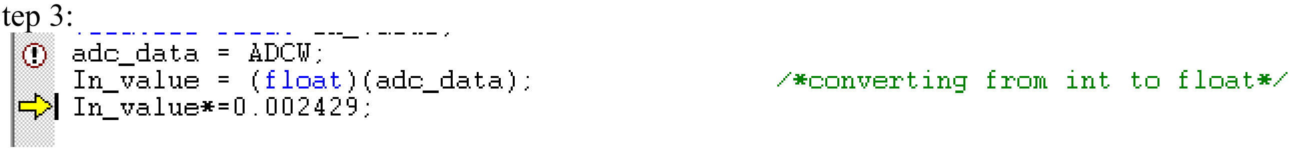
```
volatile float In_value;
adc_data = ADCW;
In_value = (float)(adc_data);                /*converting from int to float*/
In_value*=0.002429;
```

The add watch for step 2:



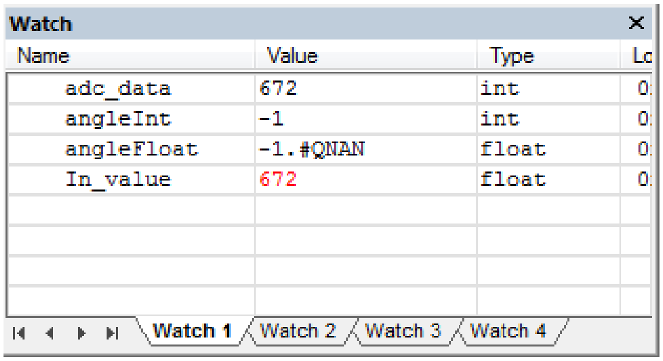
Name	Value	Type	Lc
adc_data	672	int	0:
angleInt	-1	int	0:
angleFloat	-1.#QNAN	float	0:
In_value	0	float	0:

Step 3:



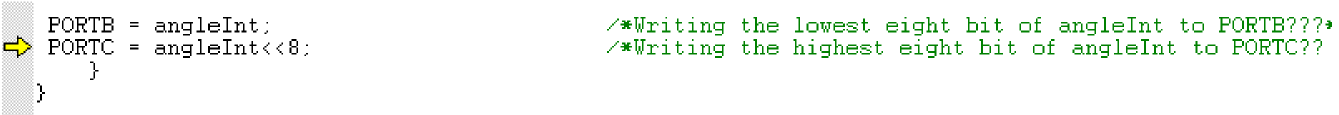
```
adc_data = ADCW;
In_value = (float)(adc_data);                /*converting from int to float*/
In_value*=0.002429;
```

The add watch for step 3:



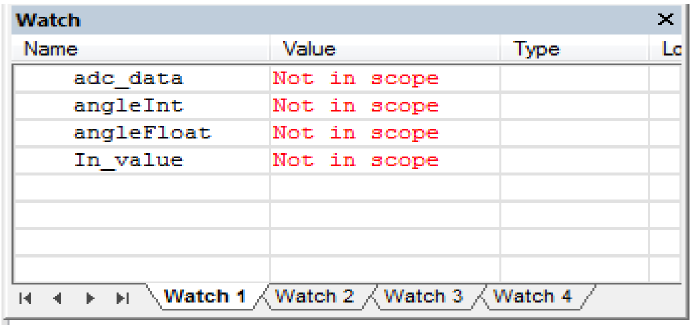
Name	Value	Type	Lc
adc_data	672	int	0:
angleInt	-1	int	0:
angleFloat	-1.#QNAN	float	0:
In_value	672	float	0:

Step 4: (in this step the debugger just jump over the if-statement)



```
PORTB = angleInt;                 /*Writing the lowest eight bit of angleInt to PORTB???*
PORTC = angleInt<<8;              /*Writing the highest eight bit of angleInt to PORTC??
    }
}
```

The add watch for step 4:



Name	Value	Type	Lc
adc_data	Not in scope		
angleInt	Not in scope		
angleFloat	Not in scope		
In_value	Not in scope		