

EE-222: Microprocessor Systems

AVR Programming in C

Instructor: Dr. Arbab Latif

Generating Time Delay in C

Time Delays in C

- You can use `for` to make time delay

```
void delay(void)
{
    volatile unsigned int i;
    for(i = 0; i < 42150; i++)
    { }
}
```

If you use for loop

- The clock frequency can change your delay duration !
- The compiler has direct effect on delay duration!

Time Delays in C

- You can use **predefined** functions of compilers to make time delays

In Atmel Studio:

First you should include:

```
#define F_CPU 8000000UL  
#include <util/delay.h>
```

and then you can use

```
_delay_us(200);    //200 microseconds  
_delay_ms(100);    //100 milliseconds
```

- It is compiler dependant

Logic Operations in C

Bit-wise logical operators

Table 7-3: Bit-wise Logic Operators for C

		AND	OR	EX-OR	Inverter
A	B	A&B	A B	A^B	Y= ~B
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1	1	
1	1	1	1	0	

1110 1111	1110 1111	
& 0000 0001	0000 0001	~ 1110 1011
-----	-----	-----
0000 0001	1110 1111	0001 0100

Bit-wise Example – (1)

- Write an AVR C program to toggle only bit 4 of Port B continuously without disturbing the rest of the pins of PORT B.

```
#include <avr/io.h>                //standard AVR header

int main(void)
{
    DDRB = 0xFF;                    //PORTB is output

    while(1)
    {
        PORTB = PORTB | 0b00010000; //set bit 4 (5th bit) of PORTB
        PORTB = PORTB & 0b11101111; //clear bit 4 (5th bit) of PORTB
    }

    return 0;
}
```

Bit-wise Example – (2)

- Write an AVR C program to monitor bit 7 of Port B. If it is 1, make bit 4 of Port B as input; otherwise, change pin 4 of Port B to output.

```
#include <avr/io.h>                                //standard AVR header

int main(void)
{
    DDRB = DDRB & 0b01111111;                      //bit 7 of Port B is input

    while (1)
    {
        if(PINB & 10000000)
            DDRB = DDRB & 0b11101111;              //bit 4 of Port B is input
        else
            DDRB = DDRB | 0b00010000;              //bit 4 of Port B is output
    }

    return 0;
}
```


Shift operations in C

- `data >> number of bits to be shifted right`
- `data << number of bits to be shifted left`

1110 0000 >> 3

0001 1100

0000 0001 <<2

0000 0100

Shift Examples

Bit Shift Statement In Hex	Resulting Value In Hex
0x01 << 1	0x02
0x02 << 1	0x04
0x04 << 1	0x08
0x01 << 2	0x04
0x01 << 3	0x08
0x08 << 1	0x10
Bit Shift Statement in Binary	Resulting Value in Binary
00000001 << 1	00000010
00000001 << 2	00000100
00001111 << 1	00011110
00001111 << 2	00111100

Setting a bit in a Byte to 1

- We can use | operator to set a bit of a byte to 1

XXXX XXXX		XXXX XXXX
0001 0000	OR	1 << 4
-----		-----
xxx1 xxxx		xxx1 xxxx

```
PORTB |= (1<<4); //set bit 4 (5th bit) of PORTB
```

Clearing a bit in a Byte to 0

- We can use `|` operator to set a bit of a byte to 1

	XXXX	XXXX				XXXX	XXXX
&	1110	1111		OR		&	$\sim(1 \ll 4)$
	-----					-----	
	xxx0	xxxx				xxx0	xxxx

```
PORTB &=  $\sim(1 \ll 4)$ ; //clear bit 4 (5th bit) of PORTB
```

Checking a bit in a Byte

- We can use & operator to see if a bit in a byte is 1 or 0

	XXXX XXXX			XXXX XXXX
&	0001 0000	OR	&	(1 << 4)
	-----			-----
	000x 0000			00x0 0000

```
if( ((PINC & (1<<5)) != 0) //check bit 5 (6th bit)
```

Bit-wise Example: Using Compound Assignment Operator

- Write an AVR C program to toggle only bit 4 of Port B continuously without disturbing the rest of the pins of PORT B.

```
#include <avr/io.h>           //standard AVR header
int main(void)
{
    DDRB &= DDRB & 0b11011111; //bit 5 of Port B is input
    while (1)
    {
        if(PINB & 0b00100000)
            DDRB &= 0b11101111; //bit 4 of Port B is input
        else
            DDRB |= 0b00010000;  //bit 4 of Port B is output
    }
    return 0;
}
```

Other Examples

- See carefully 7-15 – 7-20

Bit-wise Shift Operation: Example

```
#include <avr/io.h>                                //standard AVR header

int main(void)
{
    DDRB = DDRB & 0b01111111;                        //bit 7 of Port B is input

    while (1)
    {
        if(PINB & 10000000)
            DDRB = DDRB & 0b11101111;                //bit 4 of Port B is input
        else
            DDRB = DDRB | 0b00010000;                //bit 4 of Port B is output
    }

    return 0;
}
```

```
#include <avr/io.h>                                //standard AVR header

int main(void)
{
    DDRB = DDRB & ~(1<<7);                            //bit 7 of Port B is input

    while (1)
    {
        if(PINB & (1<<7))
            DDRB = DDRB & ~(1<<4);                    //bit 4 of Port B is input
        else
            DDRB = DDRB | (1<<4);                      //bit 4 of Port B is output
    }

    return 0;
}
```


Recommended Reading

- The AVR Microcontroller and Embedded Systems: Using Assembly and C by Mazidi et al., Prentice Hall
 - Chapter-7
 - Make sure you actually understand and run all the examples in Atmel Studio

THANK YOU

