Project #4(Pjt04_uart_intr_volatile)

Step 1

- UART0 interrupt based TX : volatile

```
void  uart_init()
{
    UBRR0H = 0x00; UBRR0L = 0x07;// 115.2K
    UCSR0A |= (1 << U2X0);
    UCSR0B |= (1 << TXEN0);
    UCSR0B |= (1 << TXCIE0);     // TX complete i
}

char  buf[64];
int   bufi, txend;

int   uart_putstart(char str[])
{
    char ch;
    strcpy(buf, str); bufi = 0; txend = 0;
    ch = buf[bufi++];

    while( !(UCSR0A & (1 << UDRE0)) )
        ;
    UDR0 = ch;
 }
```

```
main()
{
    int   i;
    uart_init();
    sei();
    for (i = 0; i < 10; i++) {
        uart_putstart( "I love you.₩n" );
          while(!txend);
        uart_putstart( "You love me.₩n" );
          while(!txend);
        uart_putstart( "He loves her.₩n" );
          while(!txend);
    }
    uart_putstart( "End !!!₩n" );
    while(1);
}
```

```
#include <avr/interrupt.h>
ISR(USART0_TX_vect)
{
   char  ch = buf[bufi];
   if (!ch) {
       txend = 1; return;
   }
   if (ch ==  '₩n' )
       buf[bufi] =  '₩r' ;
   else
       bufi++;
   UDR0 = ch;
}
```

Project #4(Pjt04_uart_intr_volatile)

Step 2

UART0 interrupt based TX :

```
void  uart_init()
{
    UBRR0H = 0x00; UBRR0L = 0x07;// 115.2K
    UCSR0A |= (1 << U2X0);
    UCSR0B |= (1 << TXEN0);
    UCSR0B |= (1 << TXCIE0);     // TX complete i
}

char  buf[64];
int   volatile  bufi, txend;
int   uart_putstart(char str[])
{
    char ch;
    strcpy(buf, str); bufi = 0; txend = 0;
    ch = buf[bufi++];

    while( !(UCSR0A & (1 << UDRE0)) )
        ;
    UDR0 = ch;
 }
```

```
main()
{
    int   i;
    uart_init();
    sei();
    for (i = 0; i < 10; i++) {
        uart_putstart( "I love you.₩n" );
          while(!txend);
        uart_putstart( "You love me.₩n" );
          while(!txend);
        uart_putstart( "He loves her.₩n" );
          while(!txend);
    }
    uart_putstart( "End !!!₩n" );
    while(1);
}
```

```
#include <avr/interrupt.h>
ISR(USART0_TX_vect)
{
   char  ch = buf[bufi];
   if (!ch) {
       txend = 1; return;
   }
   if (ch ==  '₩n' )
       buf[bufi] =  '₩r' ;
   else
       bufi++;
   UDR0 = ch;
}
```

Project #4(Pjt04_uart_intr_volatile)

Step 3

UART0 interrupt based TX :

```
void  uart_init()
{
    UBRR0H = 0x00; UBRR0L = 0x07;// 115.2K
    UCSR0A |= (1 << U2X0);
    UCSR0B |= (1 << TXEN0);
    UCSR0B |= (1 << TXCIE0);      // TX complete i
}

char  buf[64];
int   volatile  bufi, txend = 1;

int   uart_putstart(char str[])
{
    char ch;

    while(!txend);

    strcpy(buf, str); bufi = 0, txend = 0;
    ch = buf[bufi++];
    while( !(UCSR0A & (1 << UDRE0)) )
        ;
    UDR0 = ch;
 }
```

```
main()
{
    int  i;
    uart_init();
    sei();
    for (i = 0; i < 10; i++) {
        uart_putstart( "I love you.₩n" );
        uart_putstart( "You love me.₩n" );
        uart_putstart( "He loves her.₩n" );
    }
    uart_putstart( "End !!!₩n" );
    while(1);
}
```

```
#include <avr/interrupt.h>
ISR(USART0_TX_vect)
{
   char  ch = buf[bufi];
   if (!ch) {
       txend = 1; return;
   }
   if (ch ==  '₩n' )
       buf[bufi] =  '₩r' ;
   else
       bufi++;
   UDR0 = ch;
}
```