

# Enkla in- och utmatningsrutiner med USART

```
void _outchar( char c )
{
    while (( USART1->sr & (1<<7) )==0);
    USART1->dr = (unsigned short) c;
}

char _tstchar( void )
{
    if( (USART1->sr & (1<<5) )==0)
        return 0;
    return (char) USART1->dr;
}

char _inchar( void )
{
    char c;
    while ( (c=_tstchar()) ==0);
    return c;
}
```

```
typedef struct tag_usart
{
    volatile unsigned short sr;
    volatile unsigned short Unused0;
    volatile unsigned short dr;
    volatile unsigned short Unused1;
    volatile unsigned short brr;
    volatile unsigned short Unused2;
    volatile unsigned short cr1;
    volatile unsigned short Unused3;
    volatile unsigned short cr2;
    volatile unsigned short Unused4;
    volatile unsigned short cr3;
    volatile unsigned short Unused5;
    volatile unsigned short gtpr;
} USART;
#define USART1 ((USART *) 0x40011000)

void main(void)
{
    unsigned char c;
    while(1){
        c = _inchar();
        _outchar( c );
    }
}
```

# USART med avbrott

Deklarationer:

```
typedef struct tag_usart
{
    volatile unsigned short sr;
    volatile unsigned short Unused0;
    volatile unsigned short dr;
    volatile unsigned short Unused1;
    volatile unsigned short brr;
    volatile unsigned short Unused2;
    volatile unsigned short cr1;
    volatile unsigned short Unused3;
    volatile unsigned short cr2;
    volatile unsigned short Unused4;
    volatile unsigned short cr3;
    volatile unsigned short Unused5;
    volatile unsigned short gtprr;
} USART;
```

Definitioner:

```
#define USART1 ((USART *) 0x40011000)

#define USART1_IRQVEC ((void (**)(void) ) 0x2001C0D4 )

#define UE (1<<13)
#define TXEIE (1<<7)
#define TCIE (1<<6)
#define RXNEIE (1<<5)
#define TE (1<<3)
#define RE (1<<2)
#define TXE (1<<7)
#define RXNE (1<<5)

#define NVIC_USART1_ISER 0xE000E104
#define NVIC_USART1_IRQ_BPOS (1<<5)
```

# USART1 avbrott (se Quick Guide)

|    |          |        |                         |             |
|----|----------|--------|-------------------------|-------------|
| 36 | settable | SPI2   | SPI2 global interrupt   | 0x0000 00D0 |
| 37 | settable | USART1 | USART1 global interrupt | 0x0000 00D4 |
| 38 | settable | USART2 | USART2 global interrupt | 0x0000 00D8 |

IRQ-nummer=37:

$37/32 = 1 \rightarrow \text{ISER1}$

$37\%32 = 5 \rightarrow (1<<5)$

```
#define USART1_IRQVEC 0x2001C0D4
```

```
#define NVIC_ISART1_ISER 0xE000E104
#define NVIC_USART1_IRQ_BPOS (1<<5)
```

0xE000E100

0xE000E104

0xE000E108

| offset | 31            | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16            | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0          | mnemonic |
|--------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------------|----------|
| 0x000  | SETENA[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | NVIC_ISER0 |          |
| 0x004  | SETENA[63:32] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | NVIC_ISER1 |          |
| 0x008  | Reservat      |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SETENA[80:64] |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | NVIC_ISER2 |          |
|        | CLEARNA[31:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |            |          |

# USART med avbrott

```
char inbuf, outbuf;
void      usart_init( void )
{
    *((void (**)(void) ) USART1_IRQVEC ) = usart_irq_routine;
    inbuf= 0;
    *((unsigned int *) NVIC_USART1_ISER) |= NVIC_USART1_IRQ_BPOS;
    USART1->brr = 0x2D9;
    USART1->cr3 = 0;
    USART1->cr2 = 0;
    USART1->cr1 = UE | RXNEIE | TE | RE;
}
```

```
void      usart_irq_routine( void )
{
    if( USART1->sr & RXNE)
        inbuf = (char) USART1->dr;

    if(( USART1->cr1 & TXEIE)&&(USART1->sr & TXE))
    {
        USART1->dr = (unsigned short) outbuf;
        USART1->cr1 &= ~TXEIE;
    }
}
```

```
char      usart_tstchar ( void )
{
    char    c = inbuf;
    inbuf = 0;
    return c;
}
void      usart_outchar( char c )
{
    outbuf = c;
    USART1->cr1 |= TXEIE;
}
```