```
/***********************************/
/**** Software SPI Declaration *****/
/***********************************/

#include <avr/io.h>

/***** Pin Definitions *************/
/***** Depends on the AVR **********/

#define SMISO       PB6
#define SMOSI       PB5
#define SSCK        PB7
#define SCSN        PB1

#define SMOSI_PORT  PORTB
#define SMOSI_DDR   DDRB

#define SSCK_PORT   PORTB
#define SSCK_DDR    DDRB

#define SCSN_PORT   PORTB
#define SCSN_DDR    DDRB

#define SMISO_PORT  PORTB
#define SMISO_DDR   PORTB
#define SMISO_PIN   PINB

/***** Define short commands ***************/

#define SCSN_Lo SCSN_PORT &= ~(1<<SCSN);
#define SCSN_Hi SCSN_PORT |= (1<<SCSN);

#define SSCK_Lo SSCK_PORT &= ~(1<<SSCK);
#define SSCK_Hi SSCK_PORT |= (1<<SSCK);

#define Set_SMOSI   SMOSI_PORT |= (1<<SMOSI);
#define Clear_SMOSI SMOSI_PORT &= ~(1<<SMOSI);

/***** Funktions ******************/

void SSPI_Init(void);
void SSPI_Write_byte(uint8_t);
uint8_t SSPI_Read_byte(uint8_t);
```

```
/*********************************/
/******** Software SPI ***********/
/*********************************/

#include "SoftwareSPI_Dec.h"
#include <stdint.h>
#include <util/delay.h>

/******* SSPI Init **************/

void SSPI_Init(void)
{

    SMISO_DDR &= ~(1<<SMISO);                   // Set SMISO as Input
    SMOSI_DDR |= (1<<SMOSI);
    SSCK_DDR |= (1<<SSCK);
    SCSN_DDR |= (1<<SCSN);                      // Set SMOSI, SSCK and SCSN as Output

    SMISO_PORT |= (1<<SMISO);                   // Set Pullupresistor

    SSCK_Lo;
    SCSN_Hi;
}


/******* SSPI Write **************/

void SSPI_Write_byte(uint8_t Byte)
{
    uint8_t bits = 0x80;            // MSB first (LSB first => 0x01)

    while(bits)
    {
        if( Byte & bits )
        {
            Set_SMOSI;
        }
        else
        {
            Clear_SMOSI;
        }
        _delay_us(10);
        SSCK_Hi;
        _delay_us(10);
        SSCK_Lo;
        bits = bits>>1;             // schiebt nach links bis 1 rausfaellt (LSB fist <<)
    }
}

/******* SSPI Read **************/

uint8_t SSPI_Read_byte(uint8_t Byte)
{
    uint8_t bits = 0x80;            // MSB first (LSB first => 0x01)
    unsigned char data_receive = 0x00;

    while(bits)
    {
        if( Byte & bits )
        {
            Set_SMOSI;
        }
        else
        {
```

```
            Clear_SMOSI;
        }
        _delay_us(10);
        SSCK_Hi;
        _delay_us(10);
        if(SMISO_PIN & (1<<SMISO))
        {
            data_receive |= bits;

        }
        SSCK_Lo;
        bits = bits>>1;             // schiebt nach links bis 1 rausfaellt (LSB fist <<)
    }
    return data_receive;
}
```