SPI initialization erklärt

Diese Funktion `SPI_init()` initialisiert und konfiguriert die serielle Peripherie-Schnittstelle (SPI) eines AVR-Mikrocontrollers für die Kommunikation als Master. Hier ist, was in den einzelnen Schritten passiert:

1. `SPI_DDR |= (1 << SS) | (1 << MOSI) | (1 << SCK);`
    - Setzt die Bits für Chip Select (SS), Master Out Slave In (MOSI) und Serial Clock (SCK) in dem SPI-Datenrichtungsregister (`SPI_DDR`), um diese Pins als Ausgänge zu konfigurieren.
2. `SPCR = (1 << SPE) | (1 << MSTR);`
    - Setzt das SPE-Bit (SPI Enable), um die SPI-Schnittstelle zu aktivieren.
    - Setzt das MSTR-Bit (Master Mode), um den Mikrocontroller als SPI-Master zu konfigurieren.
    - Die Zeile `| (1 << SPR1) | (1 << SPR0);` ist auskommentiert, würde aber die SPI-Taktfrequenz auf fosc/128 setzen (sehr langsame Übertragungsrate).
3. `SPSR |= 0x1;` ist ebenfalls auskommentiert, würde aber die SPI-Taktfrequenz auf fosc/16 setzen.

Zusammengefasst konfiguriert diese Funktion die SPI-Schnittstelle wie folgt:

- SPI aktiviert
- Mikrocontroller als Master konfiguriert
- Chip Select (SS), MOSI und SCK als Ausgänge
- SPI-Taktfrequenz auf fosc/4 (volle CPU-Taktfrequenz geteilt durch 4)

Die Konfiguration der Taktfrequenz ist auskommentiert. Bei der Standardeinstellung ohne die SPR-Bits gesetzt wird die maximale Taktfrequenz von fosc/4 verwendet.

Diese Initialisierung ist typisch für die SPI-Master-Konfiguration, bei der der Mikrocontroller die Steuerung der SPI-Kommunikation übernimmt und Daten an einen oder mehrere SPI-Slave-Geräte sendet oder von diesen empfängt.