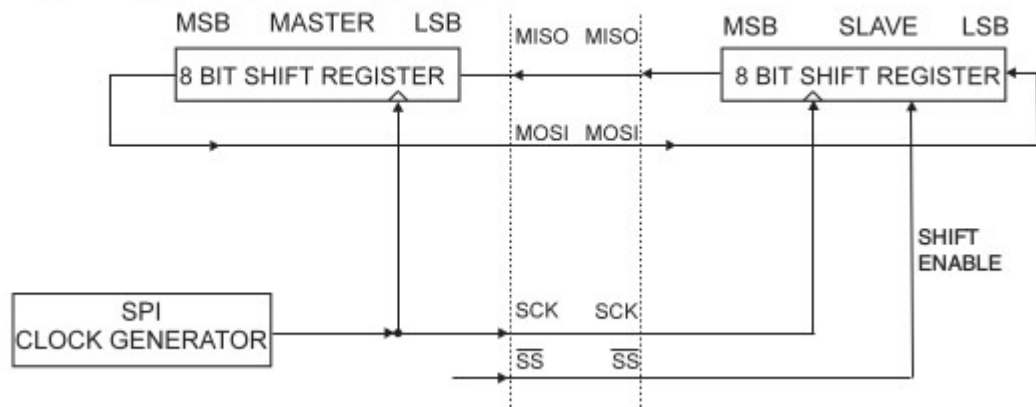


SPI Bus

Datenblatt 177

Verbindung zwischen Master und Slave

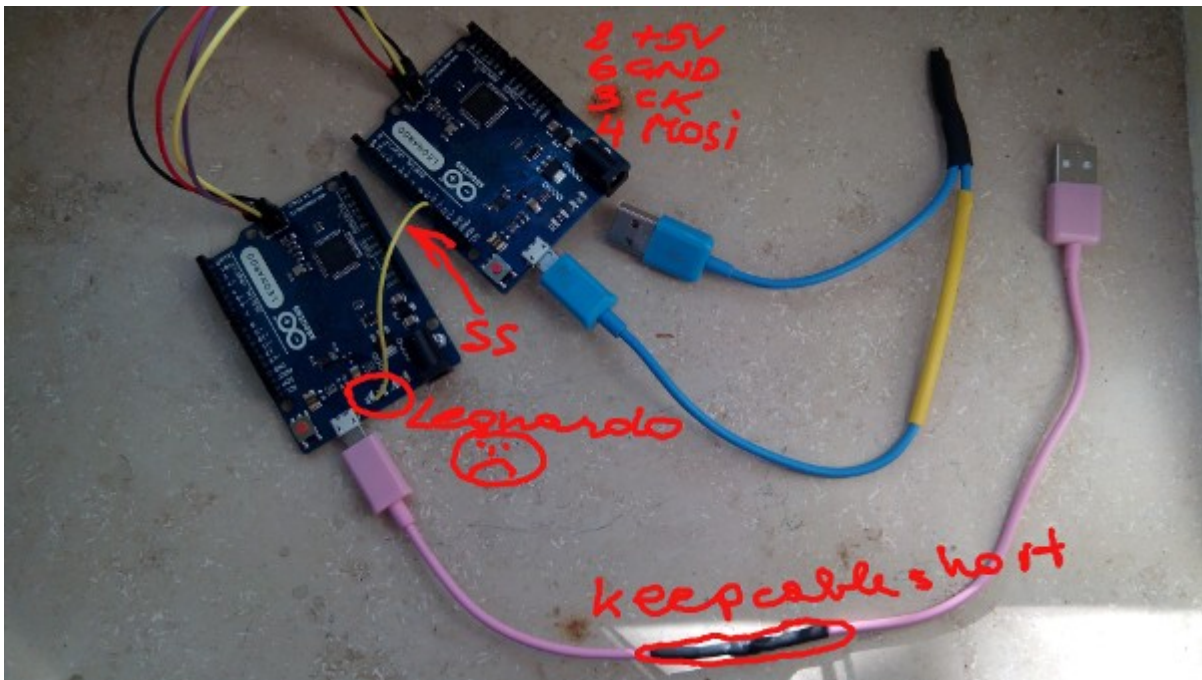
Figure 17-2. SPI Master-slave Interconnection



Grundlagen

- Master
 - erzeugt den Takt und gibt seine Daten am MOSI Pin aus
 - SS (slave select not) Pin aktiviert den Slave, wenn SS=LOW
 - wenn SS am Master1 auf Input steht kann Master1 von einem anderen Master2 in den Slave-Modus geschickt werden
 - am Master daher SS => Output
- Slave
 - am SS-Eingang wird Slave aktiviert/deaktiviert (active Low)
- SS und Arduino Leonardo
 - beim Leonardo ist der Pin nicht herausgeführt; wenn man ihn benötigt, kann man ihn an der RxLed anlöten

Probleme mit Arduino **Leonardo**: der Pin SS steht nicht zur Verfügung. Softwarelösung funktioniert nicht, der Pin muss hardware-mäßig auf GND gelegt werden. ACHTUNG: die Lösungen, die man im Internet findet, wo der Pin SS nicht angeschlossen ist oder fix auf Null liegt, funktionieren nicht. Open SS doesn't work! (initialized on rising edge of SS)



Slave

```

/* kner 2016
 * 6 Pin SPI connector on Leonardo: 1 .. Miso 3 .. SCK 5 .. RESET 2 ... +5V
4 .. MOSI 6 .. GND
 * UNO Pin 10 SS !
 * SS may stay open, must not be connected to GND; reception stops when SS=HIGH
 *
 * minimalistic Testprogramm receives 1,2,3 usw. via spi
 * dont forget to connect to UNO Pin 10 SS
 */

#include <SPI.h>

void setup (void)
{
  Serial.begin (9600); // debugging

  // have to send on master in, *slave out*
  // turn on SPI in slave mode
  SPCR |= bit(SPE);
  // have to send on master in, *slave out*
  pinMode(MISO, OUTPUT);
}

void loop (void)
{
  Serial.print("los");

  while(1){

    while(!(SPSR & (1<<SPIF)));
    int x = SPDR;
    Serial.println(x, HEX);
  }
}

```

Master

```
/* kner 2016
 * 6 Pin SPI connector on Leonardo: 1 .. Miso 3 .. SCK 5 .. RESET 2 ... +5V
4 .. MOSI 6 .. GND
 * LEONARDO R3
 *
 * minimalistic Testprogramm sends 1,2,3 usw. to spi
 */

#include <SPI.h>
#define PIN_SS 13
void setup (void)
{
  pinMode(PIN_SS, OUTPUT);
}

char i=0;
void loop (void)
{
  SPI.begin ();
  while (1){ //loop
    digitalWrite(PIN_SS, LOW);
    SPI.transfer (i++);
    digitalWrite(PIN_SS, HIGH);
    delay(100);
  }
}
```