Serial Peripheral Interface (SPI)

full-duplex, three-wire protocol

synchronous transfers

MaEvArM as master or slave

programmable bit rate

end of transmission interrupt
Serial Peripheral Interface

Communication cycle when pulling low the Slave Select SS pin of the desired Slave. Master and Slave prepare the data to be sent in their respective shift Registers, and the Master generates the required clock pulses on the SCK line to interchange data. Data is always shifted from Master to Slave on the Master Out – Slave In, MOSI, line, and from Slave to Master on the Master In – Slave Out, MISO, line. After each data packet, the Master will synchronize the Slave by pulling high the Slave Select, SS, line.

When configured as a Master, the SPI interface has no automatic control of the SS line. This must be handled by user software before communication can start. When this is done, writing a byte to the SPI Data Register starts the SPI clock generator, and the hardware shifts the eight bits into the Slave. After shifting one byte, the SPI clock generator stops, setting the end of Transmission Flag (SPIF). If the SPI Interrupt Enable bit (SPIE) in the SPCR Register is set, an interrupt is requested. The Master may continue to shift the next byte by writing it into SPDR, or signal the end of packet by pulling high the Slave Select, SS line. The last incoming byte will be kept in the Buffer Register for later use.

When configured as a Slave, the SPI interface will remain sleeping with MISO tri-stated as long as the SS pin is driven high. In this state, software may update the contents of the SPI Data Register, SPDR, but the data will not be shifted out by incoming clock pulses on the SCK pin until the SS pin is driven low. As one byte has been completely shifted, the end of Transmission Flag, SPIF is set. If the SPI Interrupt Enable bit, SPIE, in the SPCR Register is set, an interrupt is requested. The Slave may continue to place new data to be sent into SPDR before reading the incoming data. The last incoming byte will be kept in the Buffer Register for later use.

Figure 17-2. SPI Master-slave Interconnection

The system is single buffered in the transmit direction and double buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received character must be read from the SPI Data Register before the next character has been completely shifted in. Otherwise, the first byte is lost.

In SPI Slave mode, the control logic will sample the incoming signal of the SCK pin. To ensure correct sampling of the clock signal, the frequency of the SPI clock should never exceed fosc/4.
Serial Peripheral Interface: Setup
(master mode)

// disable power reduction to allow SPI:
clear(PRR0, PRSPI);

// MOSI, SCLK, and SS as output:
DDRB |= (1<<DDRB2) | (1<<DDRB1) | (1<<DDRB0);

// Enable SPI, Master mode, clock rate = sys_clk/64:
SPCR = (1<<SPE) | (1<<MSTR) | (1<<SPR1) | (1<<SPR0);
Serial Peripheral Interface: Transmission

```c
char outgoing = 0xAA;
char incoming;

// send SS low to begin transmission
clear(PORTB,0);

// send a byte to the slave
SPDR = outgoing;

// wait for transmission to finish
while(!check(SPSR,SPIF));

// read the byte returned from the slave
incoming = SPDR;

// send SS high to end transmission
set(PORTB,0);
```
Two-Wire Interface (TWI, I2C)

- simplex, two-wire protocol
- MaEvArM as master or slave
- up to 128 connected devices
Two-Wire Interface

The 2-wire Serial Interface (TWI) is ideally suited for typical microcontroller applications. The TWI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (SCL) and one for data (SDA). The only external hardware needed to implement the bus is a single pull-up resistor for each of the TWI bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are in place. Noise suppression circuitry rejects spikes on bus lines, and slew-rate limited output drivers ensure predictable propagation delays.

**TWI Terminology**

Table 20-1.

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Description</th>
</tr>
</thead>
</table>
| Master                       | The device that initiates and terminates a transmission. The Master also generates the START condition, and is used when the Master wishes to initiate a new transfer without relinquishing bus control.
| Slave                        | The device addressed by a Master.
| Transmitter                  | The device placing data on the bus.
| Receiver                     | The device reading data from the bus.

**Features**

- Fully Programmable Slave Address with General Call Support
- Noise Suppression Circuitry Rejects Spikes on Bus Lines
- Multi-master Arbitration Support
- Device can Operate as Transmitter or Receiver
- Simple Yet Powerful and Flexible Communication Interface, only two Bus Lines Needed
- Slew-rate Limited Output Drivers
- START and STOP Conditions
- Transferring Bits
- Data Transfer and Frame Format
- START condition, and is used when the Master wishes to initiate a new transmission. The transmission is initiated when the Master issues a START condition on the bus, and it is terminated when the Master issues a STOP condition, or a REPEATED START condition to initiate a new transmission.

**START and STOP Conditions**

As depicted in Figure 20-1, both bus lines are connected to the positive supply voltage through pull-up resistors. The bus drivers of all TWI-compliant devices are open-drain or open-collector. The number of devices that can be connected to the bus is only limited by the bus capacitance characteristics of the TWI is given in “SPI Timing Characteristics” on page 385. Two different sets of specifications are presented there, one relevant for bus speeds below 100 kHz, and one valid for 3.3 kHz transmission rates.

**Data Validity**

The level on a TWI bus line is generated when one or more TWI devices output a zero. A high level must be written to zero to enable the 2-wire Serial Interface. Each data bit transferred on the TWI bus is accompanied by a pulse on the clock line. The level on the SDA line is valid during the clock pulse, while the level on the SCL line is valid during clock transitions.

**Address Packet Format**

As depicted in Figure 20-4, all address packets transmitted on the TWI bus are 9 bits long, consisting of 7 address bits, one READ/WRITE control bit, and one acknowledge bit. If the READ/WRITE bit is set, a read operation will be performed, while a write operation should be performed when the READ/WRITE bit is reset. When a Slave recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL clock cycle. If the addressed Slave is busy, or for some other reason cannot service the Master's request, the SDA line should be left high in the ACK clock cycle. The Master can then transmit a STOP condition, or a REPEATED START condition to initiate a new transmission.

The MSB of the address byte is transmitted first. Slave addresses can freely be allocated by the designer, but the address 0000 000 is reserved for a general call. The following data packets will then be received by all the slaves that acknowledged the general call. The general call address followed by a Read bit is meaningless, as this would cause contention if several slaves started transmitting different data.