Serielle Zweidraht-Schnittstelle

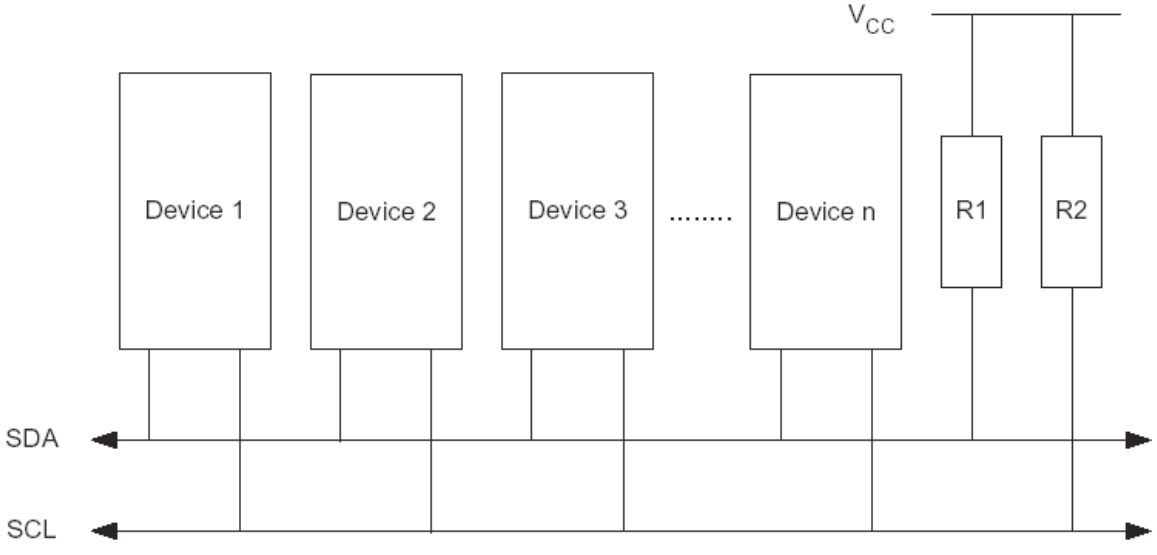
Merkmale

- einfache noch mächtige und flexible Kommunikationsschnittstelle, nur zwei erforderliche Busleitungen
- unterstützte sowohl Master- als auch Slaveoperationen
- Gerät kann als Sender oder Empfänger laufen
- 7-Bit Adressraum erlaubt bis zu 128 verschiedene Slaveadressen
- Multi-Masterschlichtungsunterstützung
- Bis zu 400 kHz Datenübertragungsgeschwindigkeit
- Slew-rate begrenzte Output Driver
- Noise-Suppression-Schaltung unterdrückt Spikes auf der Busleitung
- Voll programmierbare Slave-Addresse with General Call Support
- Adresserkennung bedingt Wake-up wenn der AVR im Sleep-Modus ist

Serielle Zweidraht-Schnittstelle, Bus Definition

Die serielle Zweidraht-Schnittstelle (TWI) ist idealerweise für typische Mikrocontrollerapplikationen geeignet. Das TWI-Protokoll erlaubt dem Systementwickler bis zu 128 verschiedene Geräte mittels nur zweier bidirektionaler Busleitungen zu verbinden, eine für den Takt SCL und eine für Daten SDA. Einzige benötigte externe Hardware ist für jede Busleitung ein einfacher Pull-up-Widerstand. Alle am Bus angeschlossenen Geräte besitzen eine individuelle Adresse. Mechanismen für das Lösen eines Busstreits sind im TWI Protokoll enthalten.

Figure 76. TWI Bus Interconnection



TWI Terminologie

Die folgenden Definitionen werden häufig in diesem Abschnitt benützt.

Term	Erklärung
Master	Das Gerät, das eine Übertragung initiiert und beendet. Der Master generiert auch den SCL Takt.
Slave	Das von einem Master angesprochene Gerät.
Transmitter	Das Gerät, das Daten auf den Bus stellt.
Receiver	Das Gerät welches Daten vom Bus liest.

Elektrische Verbindung

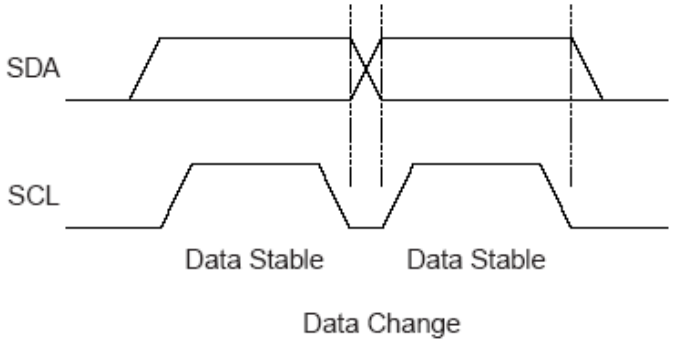
Wie in Abbildung 76 dargestellt, sind beide Busleitungen mit der positiven Versorgungsspannung durch Pullup-Widerstände verbunden. Die Bustreiber aller TWI kompatiblen Geräte besitzen Open-Collector-Ausgänge. Dies implementiert eine festverdrahtete AND-Funktion, welche für die Funktion der Schnittstelle wesentlich ist. Ein Low-Level an einer TWI Busleitung wird generiert, indem einer oder mehrere TWI-Geräte ein L ausgeben. Ein High-Level wird erzeugt, indem alle TWI-Geräte ihre Ausgänge in den tri-state Modus versetzen, dies erlaubt den Pullup-Widerständen die Leitung auf High zu ziehen.
Beachten Sie, dass alle mit dem TWI Bus verbundenen AVR Geräte eingeschaltet seien müssen, um jede Busoperation zu erlauben. Die Anzahl der am Bus anzuschließenden Geräte wird nur durch die maximale Kapazität von 400pF und den 7-Bit-Slave-Adressraum-Raum. Eine genaue Beschreibung der elektrischen Charakteristika des TWI wird in im Kapitel „Charakteristik der Seriellen Zweidraht Schnittstelle “ gegeben, auf Seite 288 (ORIGINALDATENBLATT). Dort werden zwei verschiedenen Spezifikationen gezeigt, eine, die für Bus relevant ist, beschleunigt unterhalb 100 kHz und eine gültig für Bus beschleunigt bis zu 400 kHz.

Daten Transfer und Frame Format

Transferieren von Bits

Jedes auf dem TWI Bus übertragene Datenbit wird von einem Puls auf der Taktleitung begleitet. Das Signal auf der Datenleitung muss stabil sein, wenn die Taktleitung High ist. Die einzige Ausnahme von dieser Regel dient zum Generieren von Start- und Stop Konditionen.

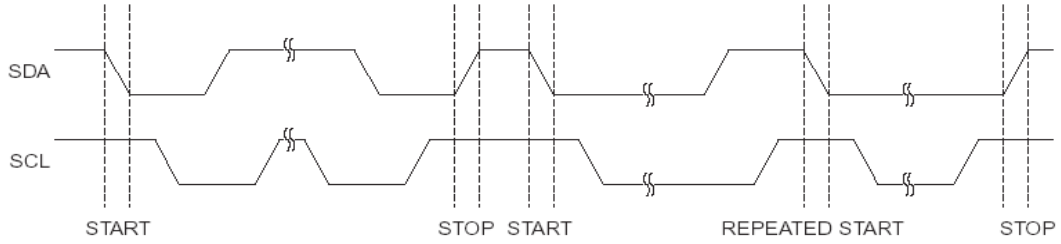
Figure 77. Data Validity



START and STOP Konditionen.

Der Master initiiert und beendet eine Datenübertragung. Die Übertragung ist initialisiert wenn der Master eine Start Kondition. auf dem Bus ausgibt, und wird beendet wenn der Master eine Stopbedingung ausgibt. Zwischen einer Start und einer Stop Kondition. ist der Bus als beschäftigt zu betrachteten, und kein anderer Master sollte versuchen, die Steuerung des Busses zu ergreifen. Ein Spezialfall tritt ein, wenn eine neue Anfangsbedingung zwischen einer Start und Stopbedingung ausgegeben wird.
Dies wird eine wiederholter Startbedingung genannt und ist gebraucht wenn der Master eine neue Übertragung initiieren will, ohne Steuerung des Busses freizugeben. Nach einem wiederholtem Start, wird der Bus bis zur nächsten Stop Kondition als beschäftigt betrachtet. Dies ist identisch dem Startverhalten, und Start wird deshalb verwendet, um beiden Starts zu beschreiben, Start und wiederholten Start, dass für den Rest dieses Datasheets , es sei denn, es ist anders beschrieben.
Wie unten dargestellt , werden die Start und Stop Konditionen**.** durch Ändern des Levels der SDA-Leitung signalisiert, während die SCL-Leitung high ist.

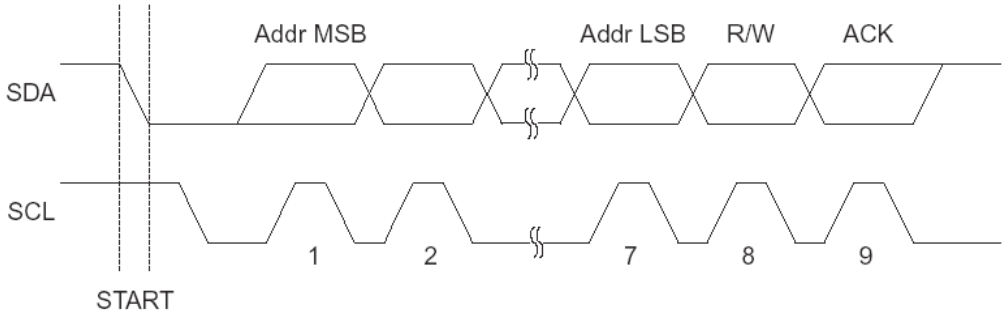
Figure 78. START, REPEATED START, and STOP Conditions



Address Paket Format

Alle Adresspakete welche über den TWI-Bus gesendet werden sind 9 Bit lang, bestehend aus 7 Adressbits, einem READ/WRITE - Controllbit und ein Acknowledgebit. Wenn das READ/WRITE - Controllbit gesetzt ist, soll eine Leseoperation ausgeführt werden, sonst sollte eine Schreiboperation ausgeführt werden. Wenn ein Slave erkennt, dass er angesprochen wird, sollte er dies durch ziehen des SDA auf Low, im neunten SCL (ACK-Takt) Zyklus bestätigen. Wenn der angesprochene Slave beschäftigt ist oder aus irgendeinem anderen Grund nicht die Anforderung des Masters bedienen kann, sollte die SDA Leitung im ACK-Takt high sein. Der Master kann eine Stop Bedingung senden, oder eine wiederholte Startbedingung um eine erneute Transmission zu initiieren. Ein Adresspaket enthält eine Slave-Adresse und ein READ oder ein WRITE Bit welches SLA+R respektive SLA+W genannt wird. Das MSB des Adress-Bytes wird zuerst gesendet. Die Slave-Adresse kann vom Entwickler frei gewählt werden, aber die Adresse 0000 000 ist für einen Generalruf reserviert. Wenn ein Generalruf ausgegeben wurde, sollten alle Slave's durch ziehen der SDA-Leitung auf Low im ACK-Takt antworten. Ein Generalruf wird gebraucht, wenn ein Meister dieselbe Nachricht an mehrere Sklaven im System senden möchte. Wenn an die Generalrufadresse, gefolgt von einem Write-Bit , auf dem Bus gesendet wird, setzen alle Slave's zum bestätigen, die SDA-Leitung im ACK-Takt auf Low. Die folgenden Datenpakete werden dann von allen den Slave's empfangen, welche den allgemeinen Anruf bestätigten. Beachten Sie, dass, das übertragen des Generalrufes gefolgt von einem Read-Bit bedeutungslos ist, da dies Kollisionen verursachen würde, wenn mehrere Slave's das senden verschiedener Daten starteten. Alle Adressen Vom Format 1111 xxx sollten für zukünftige Zwecke reserviert werden.

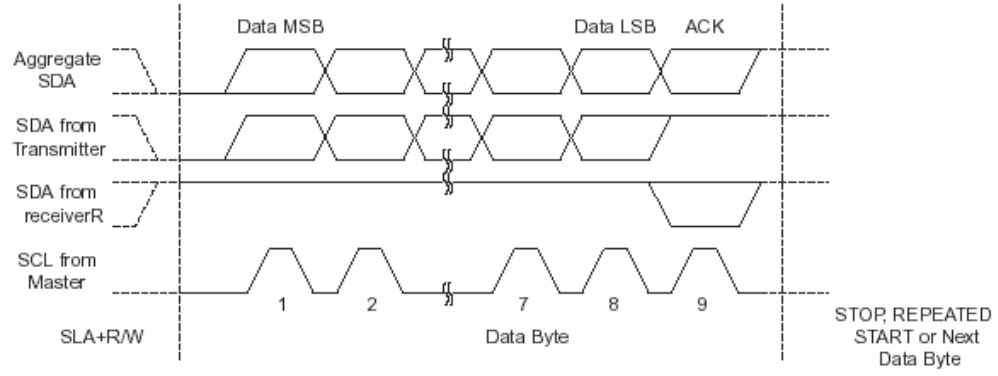
Figure 79. Address Packet Format



Daten Paket Format

Alle Datenpakete welche auf dem TWI-Bus gesendet werden sind 9 Bit lang, bestehend aus einem Data-Byte und einem Acknowledge-Bit. Während einer Datenübertragung generiert der Master de Takt und den die Start und die Stopsignale, während der Empfänger dafür verantwortlich ist, den Empfang zu bestätigen. Ein Acknowledge (ACK) wird signalisiert indem der Empfänger die SDA Leitung während des neunten SCL Taktes auf Low zieht. Wenn der Empfänger die SDA Leitung auf High belässt, wird damit ein NACK signalisiert. Wen der Empfänger das letzte Byte empfangen hat, oder aus einem anderen Grund keine mehr Bytes empfangen kann, sollte er den Sender durch Senden eines NACK's nach dem letzten Byte informieren. Das MSB des Daten Bytes wird zuerst übertragen.

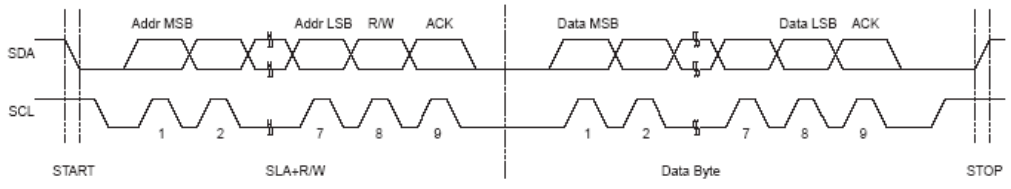
Figure 80. Data Packet Format



Kombinieren von Adress- und Daten- Paketen in einer Übertragung

Eine Übertragung beinhaltet grundlegend eine Start-Kondition, ein Slave-Adresse +R/W Bit, ein oder mehrere Datenpakete und eine Stop Kondition. Eine Leere Nachricht, bestehend aus einer START- gefolgt von einer STOP- Kondition, ist illegal. Beachten sie das die Verdrahtete AND-Funktion auf der SCL Leitung für ein Handshaking zwischen Master und Slave genutzt werden kann. Der Slave kann die Low-Dauer der SCL-Leitung verlängern , indem er die SCL-Leitung auf Low zieht. Dies kann nützlich sein wenn der takt des Masters zu schnell ist für den Slave, oder wenn der Slave extrem viel Zeit für Prozesse zwischen den Daten Übertragungen benötigt. Die Ausweitung der Low-Dauer hat keinen Einfluss auf die High-Dauer der Daten-Leitung, welche vom Master bestimmt wird. Als Konsequenz kann der Slave die TWI Datenrate reduzieren, durch Verlängerung des Taktzyklus. Figur 81 zeigt eine typische Übertragung. Beachten Sie, das mehrere Datenbytes zwischen den Slave-Adresse +R/W Bit und der Stop Kondition je nach dem von der Applikationssoftware durchgeführten Softwareprotokoll gesendet werden können.

Figure 81. Typical Data Transmission



Multi-Master Bus Systeme, Arbitration(Schlichtung) und Synchronisation

Das TWI Protokoll erlaubt Bussysteme mit mehreren Mastern. Spezielle Probleme wurden gelöst, um sicherzustellen, das Übertragungen als normal fortfahren, selbst wenn zwei oder mehr Meister zu derselben Zeit eine Übertragung initiieren.

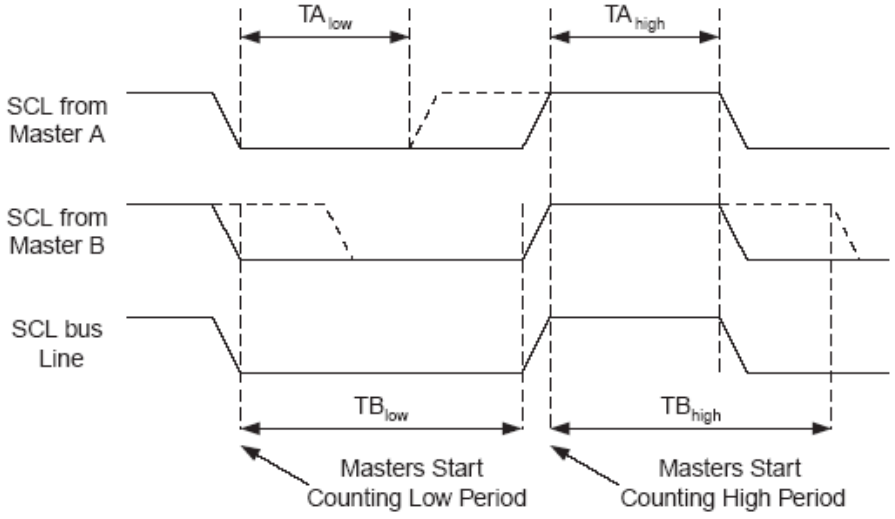
Zwei Probleme ergeben sich in Multi-Mastersystemen:

• Ein Algorithmus muss eingeführt werden der nur einem der Master erlaubt, die Übertragung zu beenden. Alle anderen Master sollten mit Übertragung aufhören, wenn sie entdecken, dass sie den Auswahlprozess verloren haben. Dieser Auswahlprozess wird Schlichtung (Arbitration) genannt. Wenn ein widerstreitender Master entdeckt, dass er den Schlichtungsprozess verloren hat, sollte er sofort umschalten zum Slave-Modus, um zu überprüfen , ob er vom gewinnenden Master angesprochen wird. Die Tatsache,

dass mehrere Master zu derselben Zeit Übertragung gestartet haben, sollte nicht durch die Slaves feststellbar sein, d.h. die Daten welche auf dem Bus übertragen werden dürfen nicht korrumpiert werden.

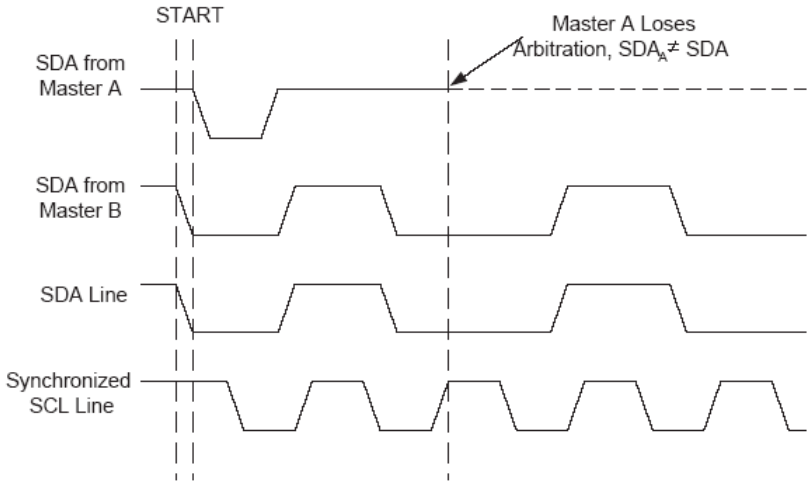
• Verschiedene Master können verschiedene SCL Frequenzen verwenden. Ein Schema muss entworfen werden, um die seriellen Takte von allen Mastern abzustimmen, um die Übertragung in einer festen Taktfolge fortfahren zu lassen. Dies erleichtert den Schlichtungsprozess. Die verdrahtete AND-Funktion der Busleitungen wird verwendet, um beide Probleme zu lösen. Die seriellen Takte aller Master wird AND verknüpft übertragen , die High-Dauer des kombinierten Taktes ist gleich der vom Master mit der kürzesten abgegebenen High-Dauer. Die folgende Low-Dauer des kombinierten Taktes ist gleich der des Masters mit der längsten Low-Dauer. Beachten Sie, dass alle Master die SCL-Leitung abhören und wirksam beginnen, ihre SCL High und Low Time-out -Dauer zu zählen, wenn die gemeinsame SCL-Leitung High beziehungsweise Low wird.

Figure 82. SCL Synchronization between Multiple Masters



Schlichtung wird von allen Mastern ausgeführt . Sie überwachen die SDA-Leitung nach dem Ausgeben von Daten stetig. Wenn der aus der SDA-Leitung gelesene Wert nicht zum Wert passt, den der Master ausgegeben hatte, hat er die Schlichtung verloren. Beachten Sie, dass ein Master die Schlichtung nur verlieren kann, wenn er ein SDA-High ausgibt, während ein anderer Master ein SDA-Low ausgibt. Der verlierende Master sollte sofort zum Slave-Modus übergehen und überprüfen, ob er vom gewinnenden Master angesprochen wird. Die SDA Leitung sollte High Bleiben, aber verlierenden Mastern wird erlaubt, ein Taktsignal bis zum Ende des gegenwärtigen Daten- oder Adresspaketes zu generieren. Schlichtung geht weiter, bis nur ein Master bleibt und dies kann viele Bits dauern. Wenn mehrere Master versuchen, denselben Slave anzusprechen, geht Schlichtung im Datenpaket weiter.

Figure 83. Arbitration between Two Masters

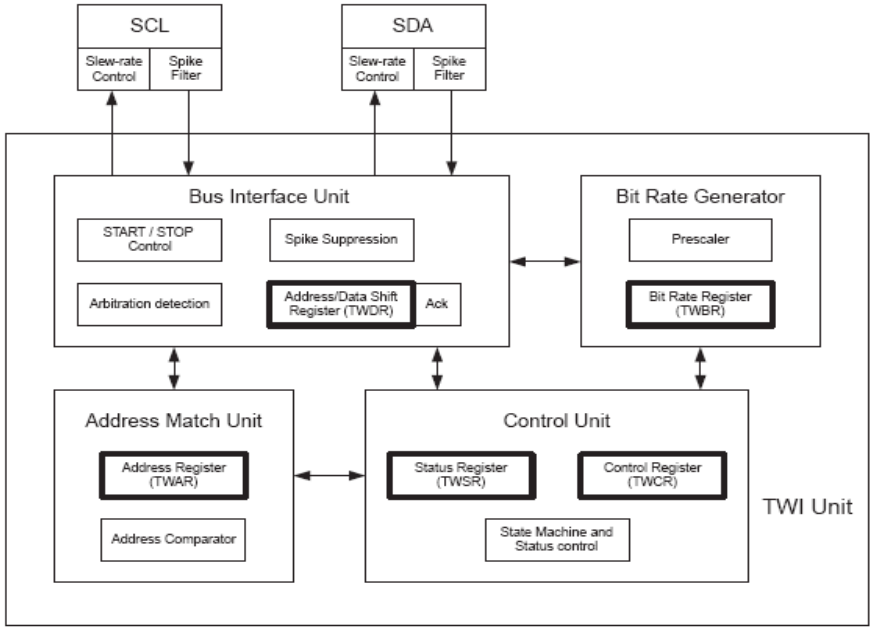


Beachten Sie, dass Schlichtung nicht erlaubt wird zwischen:
• Eine wiederholten START Kondition und einem Datenbit
• Eine Stopp Kondition und einem Datenbit
• Einer wiederholter Start- und einer Stop-Kondition
Es ist Aufgabe der Benutzersoftware, sicherzustellen, dass diese illegalen Schlichtungsbedingungen nie auftreten. Dies impliziert, dass in Multi-Mastersystemen alle Datenübertragungen dieselbe Komposition von SLA+R/W und dieselben Datenpakete verwenden müssen. Mit anderen Worten: Alle Übertragungen müssen dieselbe Anzahl von Datenpaketen enthalten, sonst ist das Ergebnis der Schlichtung undefiniert.

Überblick über das TWI Modul

Das TWI Modul besteht aus mehreren Unterbaugruppen, wie in Abbildung 84 gezeigt . Alle in einer dick eingerahmt gezeichneten Register sind durch den AVR Datenbus zugänglich.

Figure 84. Overview of the TWI Module



SCL und SDA Pins Diese Pins koppeln das AVR TWI an den Rest des MCU Systems .
Die Ausgangstreiber enthalten einen Slew-Rate-Begrenzer , um der TWI Spezifikation zu entsprechen. Die Eingangsstufenstufen enthalten eine Spike-Unterdrückungseinheit, die Spikes kürzer als 50 ns entfernt. Beachten Sie, dass das interne Pullups in den AVR Blöcken durch Setzen der Portbits ermöglicht werden kann, die den SCL und SDA Pins entsprechen, wie im I/O Portbereich erklärt. Die internen Pullups können in einigen Systemen den Bedarf für externe ersetzen.

Bit Rate Generatoreinheit Diese Einheit kontrolliert den Takt von SCL im Mastermodus.
Der SCL Takt wird von Einstellungen im TWI Bit Rate Register (TWBR) und der Prescaler- (Vorteiler-) Bits im TWI Statusregister (TWSR) kontrolliert.

Slaveoperationen hängen nicht von der Bit Rate oder Prescalereinstellungen ab, aber der CPU-takt im Slave muss mindestens 16mal höher sein, als die SCL Häufigkeit.
Beachten Sie, dass Slaves die SCL Low-Phase verlängern können, dadurch reduzieren sie die durchschnittliche TWI Busuhrperiode. SCL Frequenz wird entsprechend der folgenden Gleichung bestimmt:

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot 4^{TWPS}}$$

- TWBR = Inhalt des TWI Bit Rate Register
- TWPS = Inhalt der Prescalerbits im TWI Status Register

Notiz: TWBR sollte 10 sein oder höher wenn das TWI im Mastermodus läuft. Wenn TWBR ist, niedriger als 10 ist, kann der Master eine falsche Ausgabe auf SDA und SCL für die Erinnerung des Bytes produzieren.
Das Problem tritt auf, wenn er das TWI in Mastermodus bedient, und Start+ SLA+ R/W an einen Slave sendet (ein Slave muß nicht mit dem Bus verbunden sein, damit dies geschieht).

Die Bus-Interface-Einheit

Diese Einheit enthält das Daten- und Adressschieberegister (TWDR), Einen START/STOP Controller und die Hardware zur Schlichtungsdetektion. Das TWDR enthält die zu sendenden Adreß- oder Datenbytes oder die erhaltenen Adreß- oder Datenbytes. Zusätzlich zum 8-Bit TWDR, enthält die Busschnittstelle auch ein Register, welche das zu sendende oder zu erhaltende (N) ACK-Bit enthält. Dieses (N) ACK Register ist für die Aplikationssoftware nicht direkt zugänglich. Jedoch während des Empfangens, es kann durch Manipulieren des TWI Kontrollregisters (TWCR) H oder L gesetzt werden. Wenn in Sendermodus, kann der Wert des erhaltenen (N) ACK Stücks vom Wert im TWSR bestimmt werden. Der Start-Stop Kontroller ist für Erzeugung und Detektion von START, WIEDERHOLTER START und STOP- Konditionen verantwortlich. Der START/STOP Kontroller ist in der Lage START Und STOP Konditionen zu entdecken während die AVR MCU in einem Sleepmodus ist, welcher das Aufwecken der MCU bei einer von einem Master geschickten Adresse. Wenn das TWI eine Übertragung als Master initiiert hat, überwacht die Schlichtungsentdeckungshardware ständig die Übertragung, und versucht festzustellen, ob Schlichtung in Bearbeitung ist. Wenn das TWI eine Schlichtung verloren hat, ist das Steuerwerk informiert. Dann können Korrekturen vorgenommen werden entsprechende Statuscode werden generiert.

Adressüberwachungseinheit

Die Adressübereinstimmungsprüfungen sprechen an, wenn erhaltene Adressbytes zur 7-Bit Adresse im TWI passen, TWI Address Register (TWAR). Wenn das TWI Allgemeine Anruferkennungserlaubnis (TWGCE) Bit im TWAR ist auf eins gesetzt ist, werden alle ankommenden Adressbits auch mit der allgemeinen den Allgemeinen Anrufbits verglichen. Bei einer Adressübereinstimmung wird die Control Unit informiert, das eine entsprechende Aktion erlaubt ist. Das TWI kann oder kann nicht seine Adresse je nach Einstellungen im TWCR bestätigen. Die Adressüberwachungseinheit ist in der Lage, Adressen sogar zu vergleichen, wenn das AVR MCU im Schlafmodus ist und ermöglicht der MCU, aufzuwachen, wenn sie von einem Master angesprochen wird.

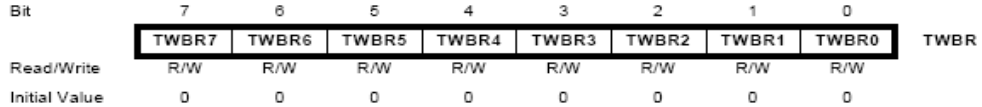
Die Control Unit

Die Control Unit überwacht den TWI Bus und generiert Antworten, die Einstellungen entsprechen denen im TWI Kontrollregister (TWCR). Wenn ein Ereignis, das die Aufmerksamkeit von der Anwendung erfordert, auf dem TWI Bus stattfindet, wird das TWI Unterbrechungsflagge (TWINT) gesetzt. Im nächsten Takt, wird das TWI Statusregister (TWSR) mit einem Statuscode aktualisiert, der das Ereignis identifiziert. Das TWSR enthält nur relevante Statusinformation, wenn die TWI Unterbrechungsflagge gesetzt ist. Zu allen anderen Zeiten enthält das TWSR einen speziellen Statuscode, der anzeigt, dass keine relevante Statusinformation verfügbar ist. Solange das TWINT Flag gesetzt ist, wird die SCL Leitung Low gehalten. Dies erlaubt der Bewerbungssoftware, seine Aufgaben vor dem Erlauben der TWI Übertragung weiterzugehen zu beenden. Das Flag ist in folgenden Situationen gesetzt.

• nach dem das TWI has transmitted a START/REPEATED START condition
• nach dem das TWI has transmitted SLA+R/W
• nach dem das TWI has transmitted an address byte
• nach dem das TWI has lost arbitration
• nach dem das TWI has been addressed by own slave address or general call
• nach dem das TWI has received a data byte
• nach eine STOP oder wiederholter START has been received while still addressed as a slave
• When a bus error has occurred due to an illegal START or STOP condition

TWI Register Beschreibung

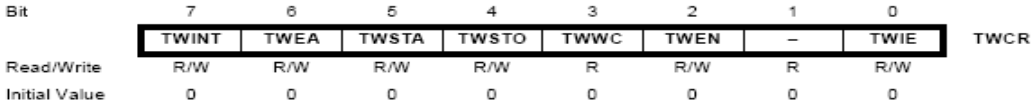
TWI Bit Rate Register – TWBR



Bit	7	6	5	4	3	2	1	0	
	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0	TWBR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7..0 – TWI Bit Rate Register
TWBR wählt den Teilungsfaktor für den Bitratengenerator. Der Bitratengenerator ist ein Frequenzsteiler, welcher den SCL Takt in den Meistermodi generiert. Siehe auch “Bit Rate Generator Unit” auf Seite 173, um Bitraten zu kalkulieren.

TWI Control Register – TWCR



Bit	7	6	5	4	3	2	1	0	
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Das TWCR wird verwendet um die Operation des TWI’s zu kontrollieren. Es wird verwendet, um dem TWI zu ermöglichen, einen Masterzugang dadurch zu initiieren, eine Startkondition auf dem Bus zu initiieren, eine Empfangsbestätigung (ack) zu generieren, um eine Stopkondition zu generieren und zu kontrollieren, und den Bus anzuhalten, während die Daten, welche werden auf den Bus zu schreiben sind, auf das TWDR geschrieben werden. Es zeigt auch einen Schreibzusammenstoß an, wenn versucht wird Daten in das TWDR zu schreiben, während das Register unzugänglich ist.
• Bit 7 - TWINT: TWI Unterbrechungsflag
Dieses BT wird von DER HARDWARE gesetzt, wenn das TWI seine gegenwärtige Aufgabe beendet hat und Softwareantwort erwartet. Wenn das I-Bit in SREG und TWIE in TWCR gesetzt sind , wird die MCU zum TWI Unterbrechungsvektor springen. Während die TWINT Flagge gesetzt ist, SCL auf Low gehalten. Das TWINT Flag muß von Software durch Hineinschreibens einer logischen Eins dazu gelöscht werden. Beachten sie, dass dieses Flag nicht automatisch von der Hardware gelöscht wird, wenn die Unterbrechungsroutine ausführt wird. Auch zu beachten ist, dass das löschen dieses Flags, die Operation des TWI’s startet, so dass alle anzusprechenden TWI Register wie TWI Address Register (TWAR), TWI Statusregister (TWSR) und TWI Datenregister an (TWDR) vollständig sein müssen, bevor dieses Flag durch Hineinschreibens einer logischen Eins dazu gelöscht werden darf

• Bit 6 – TWEA: TWI Enable Acknowledge Bit
Das TWEA Bit kontrolliert die Erzeugung des Acknowledge Bits. Wenn das TWEA Bit auf eins gesetzt ist, wird der ACK Puls auf dem TWI Bus generiert, wenn die folgenden Bedingungen erfüllt sind:

1 . die eigene Slave Adresse des Geräts ist erhalten worden.
2 . ein allgemeiner Anruf ist erhalten worden, während das TWGCE Bit im TWAR gesetzt ist.
3 . ein Datenbyte ist in Masterempfänger- oder Slave Empfängermodus erhalten worden.
Durch Schreiben einer Null in das TWEA -Bit kann das Gerät praktisch vom Zweidrahtbus abgeschaltet vorübergehend werden. Die Adresserkennung kann dann durch Schreiben einer Eins in das TWEA Bit wiederaufgenommen werden.
• Bit 5 – TWSTA: TWI START Condition Bit
die Applikation schreibt das TWSTA Bit auf eins, wenn sie wünscht, auf dem seriellen Zweidraht Bus ein Master zu werden. Die TWI Hardware überprüft, wann der Bus verfügbar ist und generiert eine Start Kondition auf dem Bus, wenn dieser frei ist. Jedoch, wenn der Bus nicht frei ist, wartet das TWI

bis eine Stop Kondition wahrgenommen wird, und generiert dann eine neue Start Kondition, um Anspruch auf den Busmasterstatus zu erheben. TWSTA muß von Software geräumt werden, wenn die Start Kondition gesendet wurde.

• Bit 4 – TWSTO: TWI STOP Condition Bit
Das TWSTO Bit auf eins im Mastermodus zu schreiben, generiert eine Stop Kondition auf dem zweidrähtigen seriellen Bus. Wenn die Stop Kondition auf dem Bus ausgeführt wird, wird das TWSTO Bit automatisch gelöscht. Im Slave Modus kann das zur Fehlerbehebung TWSTO Bit gesetzt werden. Dies generiert keine eine Stop Kondition, aber die TWI kehrt in einen nichtaufgerufenen Slave Modus zurück und gibt die SCL und SDA Leitungen hochohmig frei.

• Bit 3 – TWWC: TWI Write Collision Flag
Das TWWC Bit wird gesetzt, wenn versucht wird, auf das TWI Datenregister (TWDR) zu schreiben, während TWINT LOW ist . Dieses Flag wird durch Schreiben des TWDR Registers gelöscht, wenn TWINT High ist.

• Bit 2 – TWEN: TWI Enable Bit
Das TWEN Bit erlaubt TWI Operationen und aktiviert das TWI Interface. Wenn TWEN auf eins gesetzt ist, übernimmt das TWI die Kontrolle über die I/O Pins welche mit den SCL and SDA Pins verbunden sind, erlaubt die Slew-Ratenbegrenzung und Spikefilter. Wenn das Bit auf 0 gesetzt ist, wird das TWI abgeschaltet und alle TWI Transmissionen werden abgebrochen, ohne Rücksicht auf jede andauernde Operation.

• Bit 1 – Res: Reserved Bit
Dieses Bit ist ein reserviertes Bit und wird immer als null gelesen.

• Bit 0 – TWIE: TWI Interrupt Enable
Wenn dieses Bit auf Eins gesetzt ist und das I-Bit in SREG gesetzt ist, wird der TWI IRQ aktiviert , solange das TWINT Flag High ist.

TWI Status Register – TWSR

Bit	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	

• Bits 7..3 – TWS: TWI Status
Diese fünf Bits reflektieren den Status der TWI Logik und des zweidrähtigen seriellen Busses. Die verschiedenen Statuscodes werden später in diesem Abschnitt beschrieben. Beachten Sie, dass der vom TWSR gelesene Wert sowohl den 5-Bit Statuswert als auch den 2-Bit Prescalerwert enthält. Der Aplikation-Designer sollte die auf Null einzustellenden Prescaler -Bits maskieren, indem er sie auf null setzt, wenn er die Statusbits überprüft. Dies macht die Statusüberprüfung unabhängig von der Prescalereinstellung. Anders als sonst üblich wird in diesem Datasheet dieser Ansatz verwendet.

• Bit 2 – Res: Reserved Bit
Dieses Bit ist ein reserviertes Bit und wird immer als null gelesen.

• Bits 1..0 – TWPS: TWI Prescaler Bits
Diese Bits Können gelesen und beschrieben werden, kontrollieren die Bitrate des Präscalers.

Table 73. TWI Bit Rate Prescaler

TWPS1	TWPS0	Prescaler Value
0	0	1
0	1	4
1	0	16
1	1	64

Um dierBitrate zu berechnen, sehen Sie bitte unter “Bit Rate Generator Unit” auf Seite 173 nach. Der Wert von TWPS1 . ,0 wird bei der Gleichung verwendet.

TWI Data Register – TWDR

Bit	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

In Sendemodus, enthält das TWDR das nächste zu sendende Byte. Im Empfangsmodus, enthält das TWDR das zuletzt empfangene Byte. Es ist beschreibbar, während das TWI nicht dabei ist eine Byte zu senden. Dies ist der Fall, wenn das TWI Unterbrechungsflag (TWINT) von der Hardware gesetzt wird. Beachten Sie, dass das Datenregister nicht vom Benutzer vor dem ersten IRQ initialisiert werden kann. Die Daten in TWDR bleiben stabil, solange TWINT gesetzt ist. Während Daten heraus geschoben werden, werden auch simultan Daten über den Bus eingelesen. Das TWDR immer enthält das letzte Byte auf dem Bus, außer nach dem Erwachen aus dem Schlafmodus durch einen TWI IRQ.
In diesem Fall, ist der Inhalt von TWDR undefiniert. Im Falle einer verlorenen Busschlichtung, werden keine Daten während des Wechsels vom Master- zu Slave-Modus verloren. Handhabung des ACK Bits wird automatisch durch die TWI Logik kontrolliert. Die CPU kann nicht direkt auf das ACK Bit zugreifen.**• Bits**

7..0 – TWD: TWI Data Register
Diese acht Bits beinhalten das nächste zu sendende Datenbyte, oder das letzte empfangene Datenbyte, auf dem zweidrähtigen seriellen Bus.

TWI (Slave) Address Register– TWAR

Bit	7	6	5	4	3	2	1	0	
	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	TWAR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	0	

Das TWAR muß mit der 7-Bit Sklavenadresse geladen werden (in den sieben oberen Bits des TWAR) auf welches das TWI antwortet, wenn es als Slave- Sender oder Empfänger programmiert ist. In Multimaster Systemen muß TWAR in Mastern gesetzt werden, wenn diese als Slaves von anderen Mastern angesprochen werden können. Das LSB des TWAR wird verwendet, um das Erkennen der allgemeinen Anrufadresse ($ 00) zu ermöglichen. Es gibt einen zugehörigen Adressvergleicher, der in der erhaltenen seriellen Adresse die Slaveadresse sucht, (oder auf allgemeinen Anrufe anspricht, falls aktiviert).
Wenn eine Übereinstimmung gefunden ist, wird ein IRQ generiert.

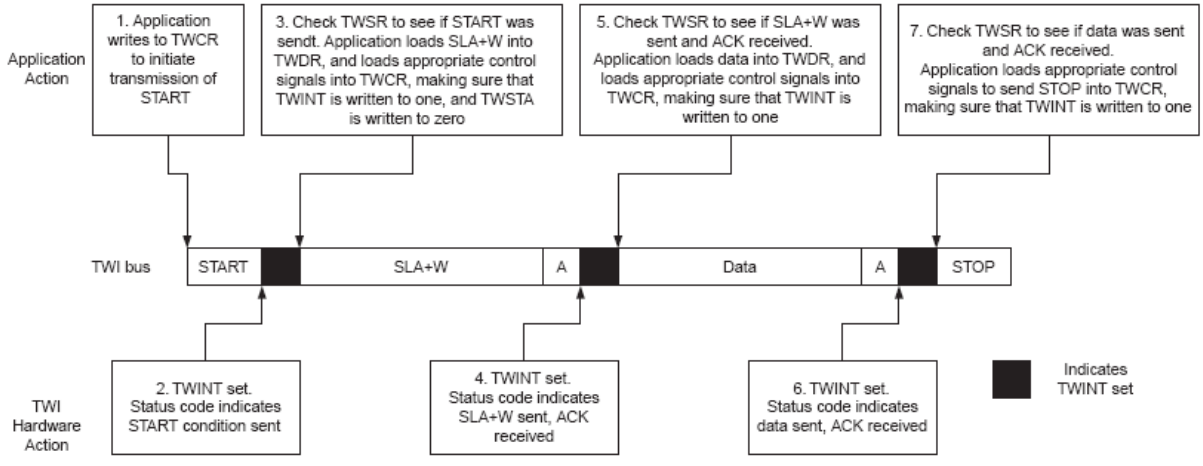
• Bits 7..1 – TWA: TWI (Slave) Address Register
Diese sieben Bits müssen die Slaveadresse der TWI-Einheit enthalten.

• Bit 0 – TWGCE: TWI General Call Recognition Enable Bit
Wenn gesetzt, erlaubt dieses Bit den Empfang des Generalrufes über den TWI-Bus.

Benutzung des TWI

Das AVR TWI ist byteorientiert, und interruptbasiert. Interrupts werden nach allem Busereignissen ausgegeben wie Empfang eines Bytes oder einer Übertragung einer Start Kondition. Weil das TWI interruptbasiert ist, ist die Aplikationssoftware frei andere Operationen während einer TWI-Byte-Übertragung durchzuführen. Beachten Sie, dass das TWI Interrupt Enable (TWIE)Bit im TWCR zusammen mit dem Global Interrupt Enable Bit im SREG der Applikation erlauben zu entscheidenden einen IRQ zu generieren, ungeachtet dessen ob das TWINT Flags gesetzt ist. Wenn das TWIE Bit gelöscht ist, muß die Applikation das TWINT Flag abfragen, um Aktionen auf dem TWI Bus zu erkennen.
Wenn das TWINT Flagge gesetzt ist, hat das TWI eine Operation beendet und erwartet Applikationsantwort. In diesem Fall enthält das TWI Statusregister (TWSR) einen Wert der den gegenwärtigen Zustands des TWI Busses anzeigt. Die Applikationssoftware kann dann entscheiden wie sich das TWI im nächsten TWI Buszyklus durch Manipulieren des TWCRs und TWDR Registers verhalten sollte.
Abbildung 85 ist ein einfaches Beispiel wie, das die Applikation an die TWI Hardware koppeln werden kann. In diesem Beispiel möchtet ein Master ein einzelnes Datenbyte an einen Slave senden. Diese Beschreibung ist ziemlich abstrakt, eine detailliertere Erklärung folgt später in diesem Abschnitt. Ein einfaches Codebeispiel, welches das gewünschte verhalten zeigt, wird auch demonstriert.

Figure 85. Interfacing the Application to the TWI in a Typical Transmission



1.
Der erste Schritt in einer TWI Übertragung ist das senden einer Start Kondition senden. Dies wird durch Eintragen eines bestimmten Werts in das TWCR erledigt, um der TWI Hardware mitzuteilen das sie eine Start Kondition senden soll. Welcher Wert einzutragen ist wird später beschrieben. Jedoch ist es wichtig, dass das TWINT Bit in dem zu schreibenden Wert gesetzt ist. Das Schreiben einer eins nach TWINT löscht das Flag. Das TWI startet keine Operation solange das TWINT Bit in TWCR ist gesetzt. Sofort nachdem die Applikation TWINT gelöscht hat initiiert das TWI die Übertragung der Start-Kondition.
2.
Wenn die Start Kondition gesendet worden ist, ist das TWINT Flag im TWCR gesetzt, und TWSR ist mit einem Statuscode aktualisiert welcher anzeigt, dass die Start Kondition erfolgreich gesendet wurde.

3 . Die Applikationssoftware sollte den Wert von TWSR jetzt prüfen, um sich zu vergewissern dass die Start Kondition erfolgreich gesendet wurde. Wenn TWSR anderes anzeigt könnte Die Applikationssoftware spezielle Maßnahmen ergreifen, wie aufrufen einer Fehlerroutine. Annehmend, dass der Statuscode erwartet wird, muß die Applikation SLA+W nach TWDR laden. Erinnern Sie sich daran, dass TWDR für beides Adresse und Daten verwendet wird . Nachdem TWDR mit dem gewünschten SLA+W geladen worden ist, muss ein bestimmter Wert auf TWCR geschrieben werden, um die TWI Hardware anzuweisen, SLA+W zu senden, welcher sich in TWDR befindet. Welcher zu schreibende Wert wird später beschrieben. Jedoch ist es wichtig, dass das TWINT Bit in den geschriebenen Wert eingesetzt ist. Das Schreiben einer eins auf TWINT löscht das Flag. Das TWI startet keine Operation, solange das TWINT Bit TWCR ist gesetzt. Sofort nachdem die Applikation TWINT gelöscht hat, wird das TWI die Übertragung des Adresspaketes initiieren
4 . Wenn das Adresspaket gesendet worden ist, wird das TWINT Flag in TWCR gesetzt und TWSR ist mit einem Statuscode aktualisiert, welcher anzeigt, dass das Adresspaket erfolgreich versandt wurde. Der Statuscode reflektiert auch ob ein Slave das Paket beantwortet oder nicht.
5 . Die Applikationssoftware sollte den Wert von TWSR jetzt prüfen, um sich zu vergewissern dass das Adresspaket erfolgreich gesendet wurde und das der Wert des ACK Stück das erwartete ist. Wenn TWSR Anderes anzeigt, könnte Applikationssoftware spezielle Maßnahmen ergreifen, wie eine Fehlerroutine aufzurufen. Annehmend ,dass der Statuscode wird erwartet, muss die Applikation ein Daten-Paket in TWDR laden. Anschließend muß ein bestimmter Wert auf TWCR geschrieben werden um die TWI Hardware anzuweisen das Daten-Paket im TWDR zu senden. Welcher zu schreibende Wert wird später beschrieben . Jedoch ist es wichtig, dass das TWINT Bit in den Wert gesetzt ist. Eine Eins nach TWINT zu schreiben, löscht das Flag. Das TWI startet keine Operation solange das TWINT Bit in TWCR gesetzt ist. Sofort nachdem löschen von TWINT durch die Applikation initiiert das TWI die Übertragung des Datenpaketes.
6 . Wenn das Datenpaket gesendet worden ist, wird das TWINT Flag in TWCR gesetzt und TWSR ist mit einem Statuscode aktualisiert, welcher anzeigt, dass das Datenpaket erfolgreich versandt wurde. Der Statuscode reflektiert auch ob ein Slave das Paket bestätigte oder nicht.
7 . Die Aplikationssoftware sollte den Wert von TWSR jetzt prüfen, um sich zu vergewissern dass das Datenpaket erfolgreich gesendet wurde, und das der Wert des ACK Bits erwartet wurde. Wenn TWSR anderes anzeigt, könnte die Applikationssoftware spezielle Maßnahmen ergreifen, wie eine Fehlerroutine aufzurufen. Annehmend dass der Status Code erwartet wird, muss die Applikation einen bestimmten Wert auf TWCR schreiben welcher die TWI Hardware anweist, eine Stopp Kondition zu senden. Welchen zu schreibenden Wert wird später beschrieben. Jedoch ist es wichtig, dass das TWINT Bit in dem Wert gesetzt ist. Eine eins auf TWINT zu schreiben, löscht das Flag. Das TWI startet nicht irgendwelche Operation, solange das TWINT Bit in TWCR gesetzt ist. Sofort nach dem Löschen von TWINT durch die Applikation initiiert das TWI die Übertragung der Stop Kondition
Beachten Sie, dass TWINT nicht gesetzt wird, nachdem eine Stop Kondition gesandt worden ist.
Obwohl dieses Beispiel einfach ist, zeigt es die Prinzipien mit allen TWI Übertragungen verbunden. Diese können wie folgt zusammengefasst werden:

• wenn das TWI eine Operation beendet hat und eine Softwareantwort erwartet , ist das TWINT Flag gesetzt. Die SCL Leitung wird auf LOW gezogen, bis TWINT gelöscht ist.

• wenn das TWINT gesetzt ist, muß der Benutzer alle TWI Register mit den für den nächsten TWI Buszyklus relevanten Werten aktualisieren. Als ein Beispiel muß TWDR mit geladen werden dem im nächsten Buszyklus zu sendendem Wert.

• nach dem aktualisieren aller TWI Register, und der Erledigung anderer anstehender Applikationssoftwareaufgaben. Währen des Schreibens in das TWCR, sollte das TWINT Bit gesetzt sein

Eine eins auf TWINT zu schreiben, löscht die Flag. Das TWI beginnt dann zu arbeiten gleich welche Operation auch immer von der TWCR Einstellung vorgegeben wurden.
Im Folgenden eine werden eine Assembler- und eine C- Implementation als Beispiel gegeben. Beachten Sie das der Code unten annimmt, dass mehrere Definitionen gemacht worden sind, zum Beispiel durch Einschließen von Include -Files.

/	**Assembler Code Beispiel**	**C Beispiel**	**Kommentare**
1	`ldi r16,(1<<TWINT)|(1<<TWSTA)|(1<<TWEN)` `out TWCR, r16`	`TWCR=(1<<TWINT)|(1<<TWSTA)|(1<<TWEN)`	Senden der START Kondition

2	`wait1:`
	`in r16,TWCR`
	`sbrs r16,TWINT`
	`rjmp wait1`
		`while (!(TWCR & (1<<TWINT)));`	Warten darauf, dass TWINT Flag gesetzt wird. Dies zeigt an dass die Start Kondition gesendet wurde.

3	`in r16,TWSR`
	`andi r16, 0xF8`
	`cpi r16, START`
	`brne ERROR`
		`if ((TWSR & 0xF8) != START)`
		`ERROR();`
			Überprüfung Wertes aus dem TWI Statusregisters. Maskieren der Prescaler-Bits. Wenn Status anders als Start gehe zu Fehler.

	`ldi r16, SLA_W`
	`out TWDR, r16`
	`ldi r16, (1<<TWINT) | (1<<TWEN)`
	`out TWCR, r16`
		`TWDR = SLA_W;`
		`TWCR = (1<<TWINT) | (1<<TWEN);`
			Lade SLA_W in das TWDR Register. Löschen des TWINT Bit in TWCR um die Übertragung zu starten.

4	`wait2:`
	`in r16,TWCR`
	`sbrs r16,TWINT`
	`rjmp wait2`
		`while (!(TWCR & (1<<TWINT)))`
		`;`
			Warten darauf ,dass TWINT Flag gesetzt wird. Dies zeigt an, dass SLA+W übertragen wurden. und ACK/NACK empfangen wurde.

5	`in r16,TWSR`
	`andi r16, 0xF8`
	`cpi r16, MT_SLA_ACK`
	`brne ERROR`
		`if ((TWSR & 0xF8) != MT_SLA_ACK)`
		`ERROR();`
			Überprüfung Wertes aus dem TWI Statusregisters. Maskieren der Prescaler-Bits. Wenn Status anders als MT_SLA_ACK gehe zu Fehler.

	`ldi r16, DATA`
	`out TWDR, r16`
	`ldi r16, (1<<TWINT) | (1<<TWEN)`
	`out TWCR, r16`
		`TWDR = DATA;`
		`TWCR = (1<<TWINT) | (1<<TWEN);`
			Lade DATA in das TWDR Register Löschen des TWINT Bit in TWCR um die Übertragung zu starten.

6	`wait3:`
	`in r16,TWCR`
	`sbrs r16,TWINT`
	`rjmp wait3`
		`while (!(TWCR & (1<<TWINT)))`
		`;`
			Warten darauf ,dass TWINT Flag gesetzt wird. Dies zeigt an, dass DATA übertragen wurden, und ACK/NACK empfangen wurde.

7	`in r16,TWSR`
	`andi r16, 0xF8`
	`cpi r16, MT_DATA_ACK`
	`brne ERROR`
		`if ((TWSR & 0xF8) != MT_DATA_ACK)`
		`ERROR();`
			Überprüfung Wertes aus dem TWI Statusregisters. Maskieren der Prescaler-Bits. Wenn Status anders als MT_DATA_ACK gehe zu Fehler.

	`ldi r16,(1<<TWINT)|(1<<TWEN)|(1<<TWSTO)` `out TWCR, r16`	`TWCR = (1<<TWINT)|(1<<TWEN)|` `(1<<TWSTO);`	Senden der STOP Kondition

Übertragungsmodi

Das TWI kann in einem von vier Major Modi laufen. Diese werden Master Transmitter (MT), beherrschen MT), Master Receiver (MR), Slave Transmitter (ST) und Slave Receiver (SR) genannt. Mehrere von diese Modi können bei derselben Applikation verwendet werden. Als ein Beispiel kann das TWI verwenden MT Modus, um Daten in einen TWI EEPROM zu schreiben, MR Modus um die Daten vom EEPROM zurück zu lesen. Wenn andere Master anwesend im System sind, könnten einige von diesen Daten zum TWI senden dann würde der SR Modus gebraucht. Es ist die Applikationssoftware, welche entscheidet, welche Modi erlaubt sind. Die folgenden Abschnitte beschreiben jeden dieser Modi. Mögliche Statuscodes sind beschrieben zusammen mit Zahlen, die Datenübertragung ausführlich darstellen, in jedem der Modi. Diese Zahlen enthalten Sie die folgenden Abkürzungen:

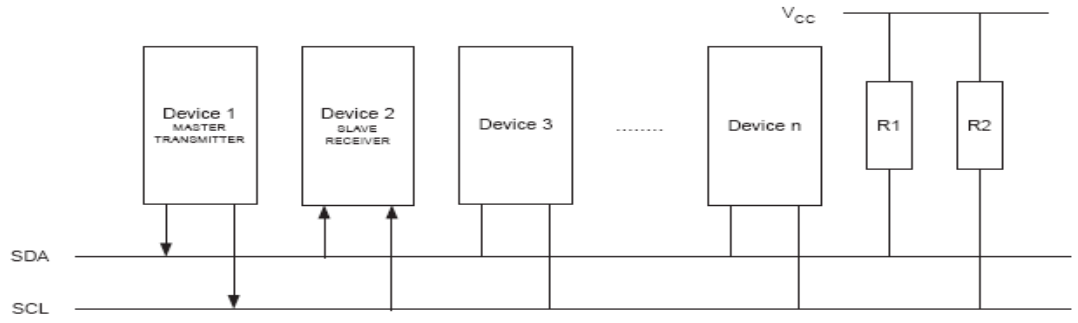
S: Start Kondition
Rs: Wiederholte Start Kondition
R: Lese Bit (high level an SDA)
W: Schreib Bit(low level an SDA)
A: Bestätigungs-Bit (low level an SDA)
/A: Nicht-Bestätigungs-Bit (high level an SDA)
Data: 8-Bit Datenbyte
P: STOP Kondition
SLA: Slave Address

In Abbildung 87 bis Abbildung 93,werden Kreise verwendet, um anzuzeigen, dass das TWINT Flag gesetzt ist. Die Nummern in den Kreisen zeigen den in TWSR mit den Prescalerstücken gehaltenen Statuscode verborgen, um auf Null zu stehen. An diesen Punkten müssen Aktionen von der Applikation unternommen werden, um die TWI Übertragung fortzusetzen, oder sie zu beenden . Die TWI Übertragung ist angehalten, bis das TWINT Flag ist von Software gelöscht wird.
Wenn das TWINT Flag gesetzt wird, wird der Statuscode in TWSR verwendet, um die entsprechende Softwareaktion auszuwählen. Für jeden Statuscode sind in den Tabellen in 74-77 die erforderlichen Softwareaktion und Details für folgende serielle Übertragung wird gegeben . Zu beachten das die Prescaler Bits zu 00 zu maskiert sind, um in diesen Tabellen auf Null zu stehen.

Master Transmitter Mode

Im Master Transmitter Modus, ein Anzahl von Datenbytes ist an einen Sklaven gesendet Empfänger (Abbildung 86). Um in den Mastermodus zu gelangen, muß eine Start Kondition gesendet sein. Das Format des folgenden Adresspaketes bestimmt ob Master Transmitter oder Master Receiver genutzt werden soll. Wenn an SLA+W gesendet wird, wird MT Modus verwendet, wenn SLA+R gesendet wird, wird der MR Modus benutzt. Alle in diesem Abschnitt erwähnten Statuscode gehen davon aus , dass die Prescaler-Bits null sind oder zu null maskiert sind, um auf Null zu stehen.

Figure 86. Data Transfer in Master Transmitter Mode



Eine Start Kondition wir gesendet durch schreiben des folgenden Wertes auf TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

TWEN muss gesetzt sein um die zweidrähtige seriellen Schnittstelle zu gestatten, TWSTA muss 1 gesetzt werden um eine Startkondition senden zu können und TWINT auf eins geschrieben werden, um das TWINT Flag zu löschen. Das TWI testet den zweidrähtigen seriellen Bus und generiert eine Start Kondition, sobald der Bus frei wird. Nachdem eine Anfangsbedingung gesendet worden ist, wird das TWINT Flag von der Hardware gesetzt, und der Statuscode in TWSR wird $ 08 sein (siehe Tabelle 74). Um in den MT Modus einzutreten, muß SLA+W gesendet werden. Dies wird durch Schreiben von SLA +W nach TWDR getan. Danach sollte das TWINT Bit (durch setzen auf 1) gelöscht werden um die Übertragung fortzusetzen. Dies wird durch Schreiben des folgenden Werts auf TWCR erreicht:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

Wenn SLA+W gesendet worden sind, und ein ACK Bit erhalten worden ist, wird TWINT wieder gesetzt, und der Status Code liegt in TWSR . Mögliche Status Codes im Mastermodus sind $ 18, $ 20 oder $ 38. Die entsprechende Aktion für jeden dieser Statuscodes ist detailliert in Tabelle 74 zu entnehmen.
Wenn SLA +W erfolgreich gesendet worden ist, sollte ein Daten-Paket gesendet werden.
Dies wird durch Schreiben des Datenbytes auf TWDR getan. TWDR kann nur dann geschrieben werden wenn TWINT high ist. Wenn nicht ist der Zugang gesperrt, und das Schreib-Kollisions-Bit (TWWC) im TWCR Register wird gesetzt. Nach dem Aktualisieren von TWDR sollte das TWINT Bit gelöscht werden (durch hineinschreiben von 1), um die Übertragung fortzusetzen. Dies wird durch Schreiben erreicht des Folgen des Wertes nach TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

Dieses Schema wird wiederholt, bis das letzte Byte gesandt worden ist und die Übertragung beendet wird, durch das Generieren einer Stop Kondition oder einer wiederholten Start Kondition. Eine Stop Kondition wird generiert durch Schreiben des folgenden Werts auf TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	1	X	1	0	X

Eine wiederholte Start Kondition wird generiert durch Schreiben des folgenden Werts auf TWCR:

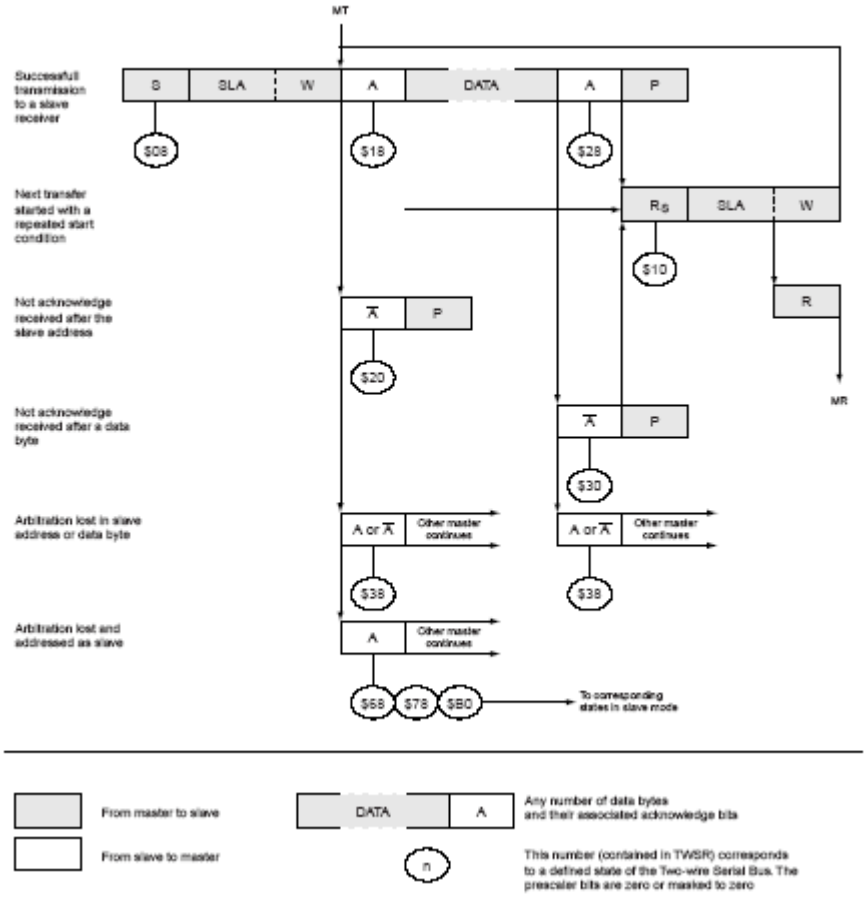
TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

Nach einer wiederholten Start Kondition (State $10) kann die zweidrähtige serielle Schnittstelle auf den selben Slave noch einzugreifen, oder auf einen anderen Slave ohne eine Stop Kondition zu senden. Wiederholter Start ermöglicht dem Master, zwischen Slaves, Master Transmitter Mode und Master-Receiver-Mode ohne die Kontrolle über die Steuerung des Busses zu verlieren.

Tabelle 74. Status verschlüsselt für Master Transmitter mode

Status Code (TWSR) Prescaler Bits sind 0	Status des Two-wire-Serial Bus und der Two-wire-Serial lnterfaceHardware	Applikationssoftwareantwort					Nächste Aktion der TWI- Hardware
		zum /vom TWDR	zum TWCR				
			STA	STO	TWINT	TWEA	
$08	Eine START-Kondition wurde übertragen	Lade SLA+W	0	0	1	X	SLA+W wird übertragen, ACK oder NACK wird empfangen
$10	Eine wiederholte START-Kondition wurde übertragen	Lade SLA+W	0	0	1	X	SLA+W wird übertragen, ACK oder NACK wird empfangen
		oder					
		Lade SLA+R	0	0	1	X	SLA+R wird übertragen, ACK oder NACK wird empfangen
$18	SLA+W wurde gesendet und wurde ACK empfangen	Lade Datenbyte	0	0	1	X	Datenbyte wird gesendet, und ACK oder NACK werden empfangen
		oder					
		Keine TWDR Aktion	1	0	1	X	Wiederholter Start wird gesendet
		oder					
		Keine TWDR Aktion	0	1	1	X	Stop wird gesendet und TWSTO Flag wird gelöscht
		oder					
		Keine TWDR Aktion	1	1	1	X	Stop Kondition, gefolgt von einer Start Kondition wird gesendet, und TWSTO Flag wird gelöscht.
$20	SLA+W wurde gesendet und NOT ACK wurde empfangen	Lade Datenbyte	0	0	1	X	Datenbyte wird gesendet, und ACK oder NACK werden empfangen
		oder					
		Keine TWDR Aktion	1	0	1	X	Wiederholter Start wird gesendet
		oder					
		Keine TWDR Aktion	0	1	1	X	Stop wird gesendet und TWSTO Flag wird gelöscht
		oder					
		Keine TWDR Aktion	1	1	1	X	Stop Kondition, gefolgt von einer Start Kondition wird gesendet, und TWSTO Flag wird gelöscht.
$28	Daten Byte wurde gesendet und ACK wurde empfangen	Lade Datenbyte	0	0	1	X	Datenbyte wird gesendet, und ACK oder NACK werden empfangen
		oder					
		Keine TWDR Aktion	1	0	1	X	Wiederholter Start wird gesendet
		oder					
		Keine TWDR Aktion	0	1	1	X	Stop wird gesendet und TWSTO Flag wird gelöscht
		oder					
		Keine TWDR Aktion	1	1	1	X	Stop Kondition, gefolgt von einer Start Kondition wird gesendet, und TWSTO Flag wird gelöscht.
$30	Daten Byte wurde gesendet und NOT ACK wurde empfangen	Lade Datenbyte	0	0	1	X	Datenbyte wird gesendet, und ACK oder NACK werden empfangen
		oder					
		Keine TWDR Aktion	1	0	1	X	Wiederholter Start wird gesendet
		oder					
		Keine TWDR Aktion	0	1	1	X	Stop wird gesendet und TWSTO Flag wird gelöscht
		oder					
		Keine TWDR Aktion	1	1	1	X	Stop Kondition, gefolgt von einer Start Kondition wird gesendet, und TWSTO Flag wird gelöscht.
$38	Schlichtung wurde während der Übertragung von SLA+W oder des Daten Bytes verloren	Keine TWDR Aktion	0	0	1	X	Zweidrahtiger serieller Bus wird freigegeben und nicht angesprochen, Slave Modus tritt ein
		oder					
		Keine TWDR Aktion	1	0	1	X	Eine START Kondition wird gesendet, wenn der Bus frei wird.

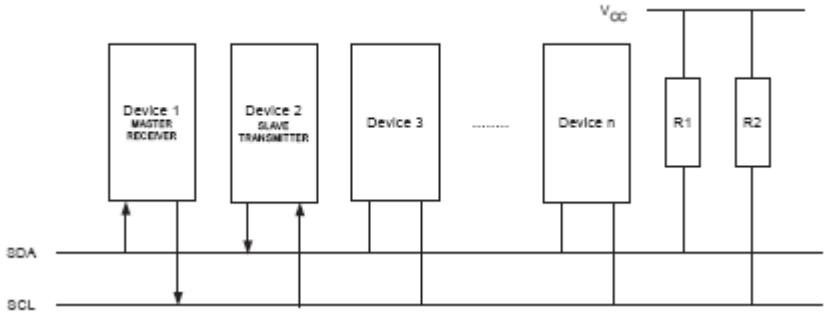
Figure 87. Formats and States in the Master Transmitter Mode



Master Receiver Mode

Im Master Receiver Modus, wird ein Anzahl von Datenbytes von einem Slave Transmitter erhalten (Abbildung 88 sehen). Um in den Mastermodus einzutreten, muß eine Start Kondition gesendet werden. Das Format der folgenden Adresspaketes bestimmt ob Master Transmitter oder Master Receiver Modus benutzt werden soll. Wenn an SLA +W gesendet wurde, wird in den MT Modus eingetreten, wenn SLA+R gesendet wurde, wird in den MR Modus wird eingetreten. Alle in diesem Abschnitt erwähnten Statuscode gehen davon aus , dass die Prescaler-Bits null sind oder zu null maskiert sind, um auf Null zu stehen.

Figure 88. Data Transfer in Master Receiver Mode



A START condition is sent by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

TWEN must be written to one to enable the Two-wire Serial Interface, TWSTA must be written to one to transmit a START condition and TWINT must be set to clear the TWINT Flag. The TWI will then test the Two-wire Serial Bus and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT Flag is set by hardware, and the status code in TWSR will be $08 (See Table 74). In order to enter MR mode, SLA+R must be transmitted. This is done by writing SLA+R to TWDR. Thereafter the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

When SLA+R have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in master mode are $38, $40, or $48. The appropriate action to be taken for each of these status codes is detailed in Table 75. Received data can be read from the TWDR Register when the TWINT Flag is set high by hardware. This scheme is repeated until the last byte has been received. After the last byte has been received, the MR should inform the ST by sending a NACK after the last received data byte. The transfer is ended by generating a STOP condition or a repeated START condition. A STOP condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	1	X	1	0	X

A REPEATED START condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

After a repeated START condition (state $10) the Two-wire Serial Interface can access the same slave again, or a new slave without transmitting a STOP condition. Repeated START enables the master to switch between slaves, Master Transmitter mode and Master Receiver mode without losing control over the bus.