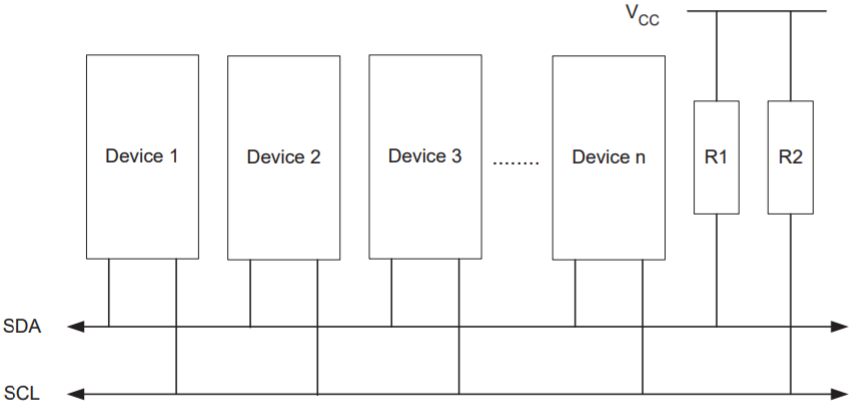
**I2C/TWI (Two-Wire Interface) Communication Protocol Notes**

Connections:

M/S Device Connections to ATMega16A

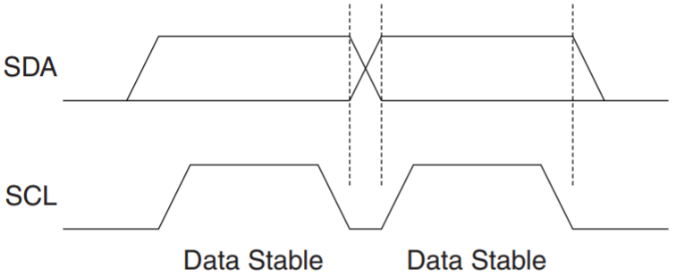


SDA (Serial Data) – Bi-directional data-transmission bus

SCL (Serial Clock) – Bi-directional clock bus

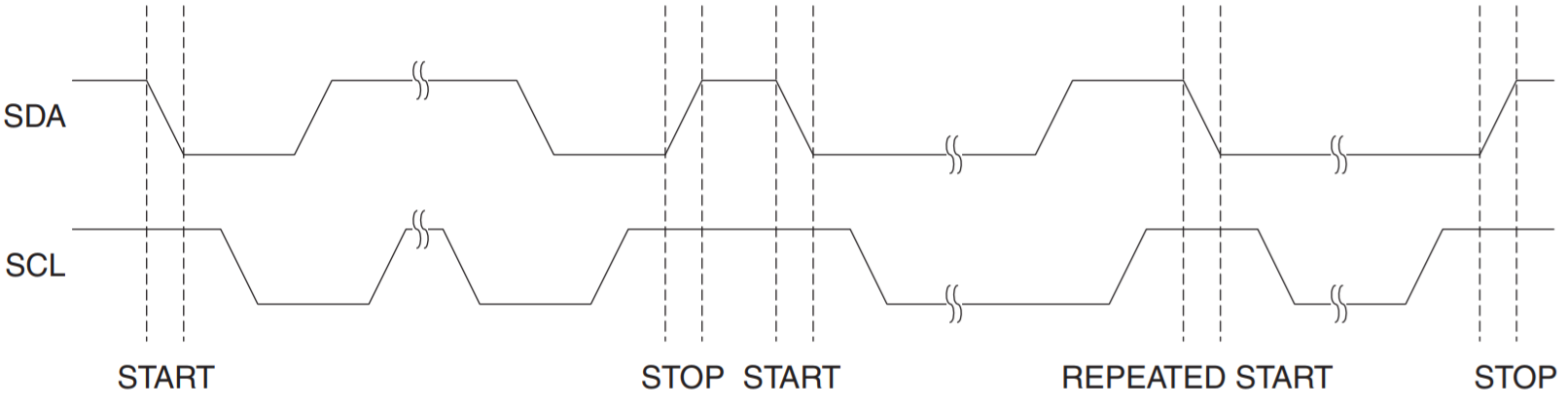
R1/R2 – Pull-up resistors

* Bus drivers of all TWI-compliant devices are open-collector (BJT) or open-drain (CMOS)
  + Buses are pulled high when transistors/FETs are opened, making buses wired-AND (all transistors/FETs need to be open for bus to drive logic high)

Data Transfer:

- Besides during START, REPEATED START, and STOP conditions, data must be stable while *SCL* is high.

START and STOP Conditions:

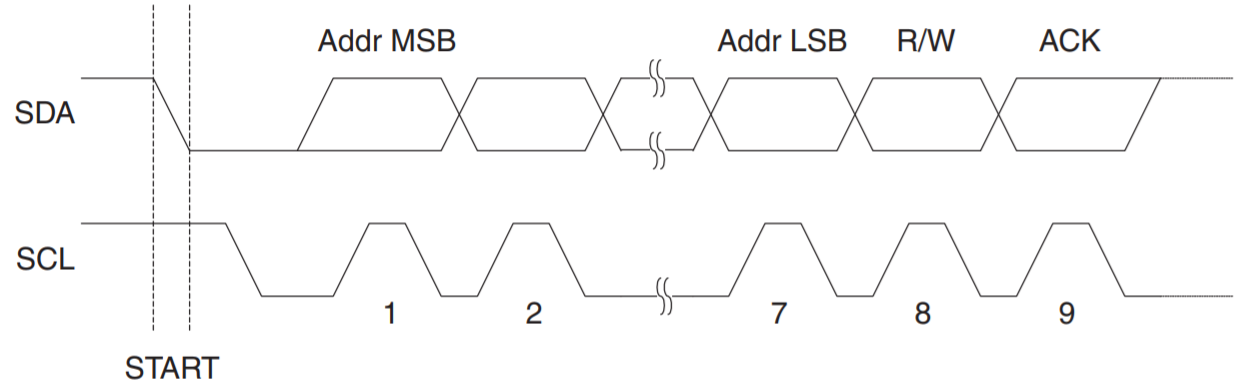


* START, REPEATED START, and STOP conditions are only issued by a Master.
* Between transmission start and stop, the bus is busy, and no other Master should try to seize control.

Condition Table [***SCL* = 1** to issue conditions]

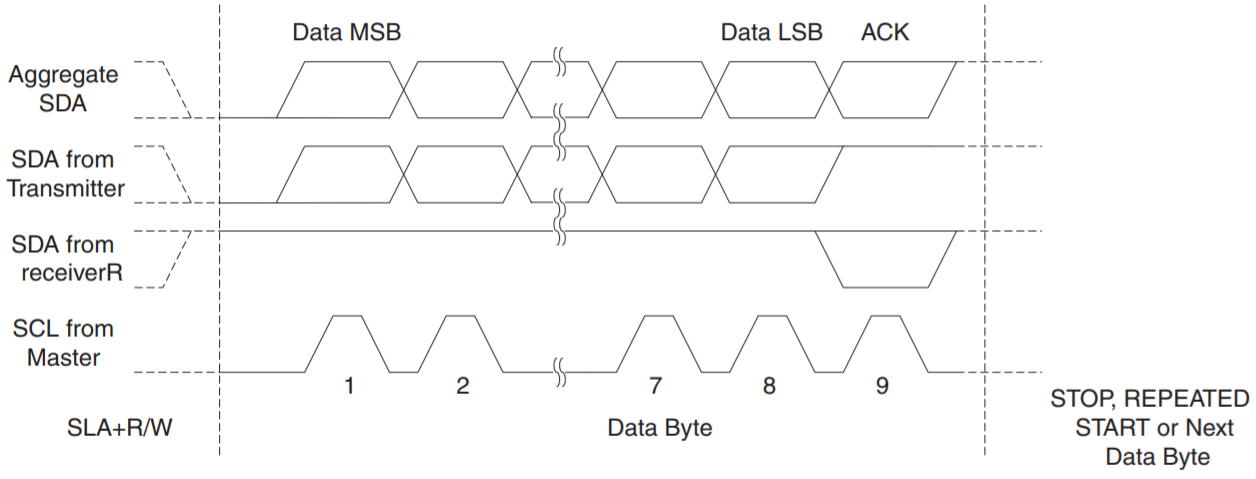
|  |  |  |
| --- | --- | --- |
| **Condition** | **Description** | **SDA** |
| START | Condition to start transmission. | 1 🡪 0 |
| REPEATED START | Condition to repeat a transmission without releasing bus control | 1 🡪 0 |
| STOP | Condition to end transmission | 0 🡪 1 |

Address Packet Format:



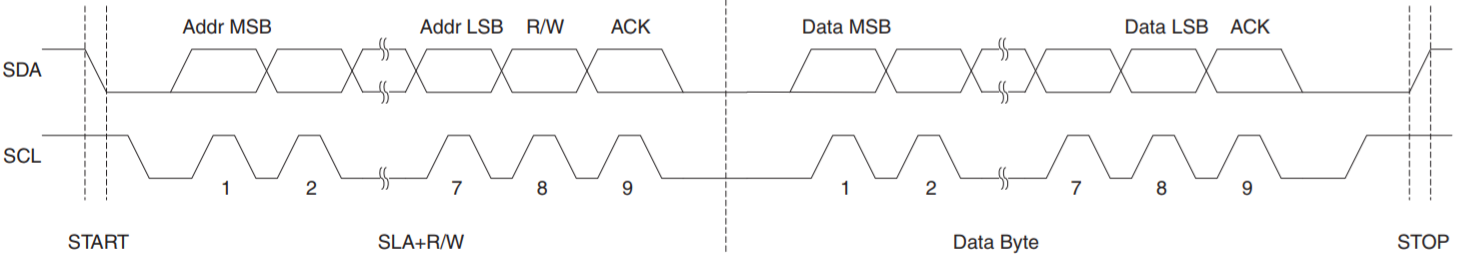
* Address Packet consists of 9 bits
  + First 7 bits indicates slave address, chosen by slave device designer
  + 8th bit is R/W (Read/Write) bit
    - R/W = 1 🡪 Read from Slave
    - R/W = 0 🡪 Write to Slave
  + **FIRST 8 bits known as SLA+R/W**
  + 9th bit is ACK (Acknowledge) bit
    - ACK = 1 🡪 Indicates Slave has not acknowledged transmission
    - ACK = 0 🡪 Indicates Slave has acknowledged transmission

Data Packet Format:



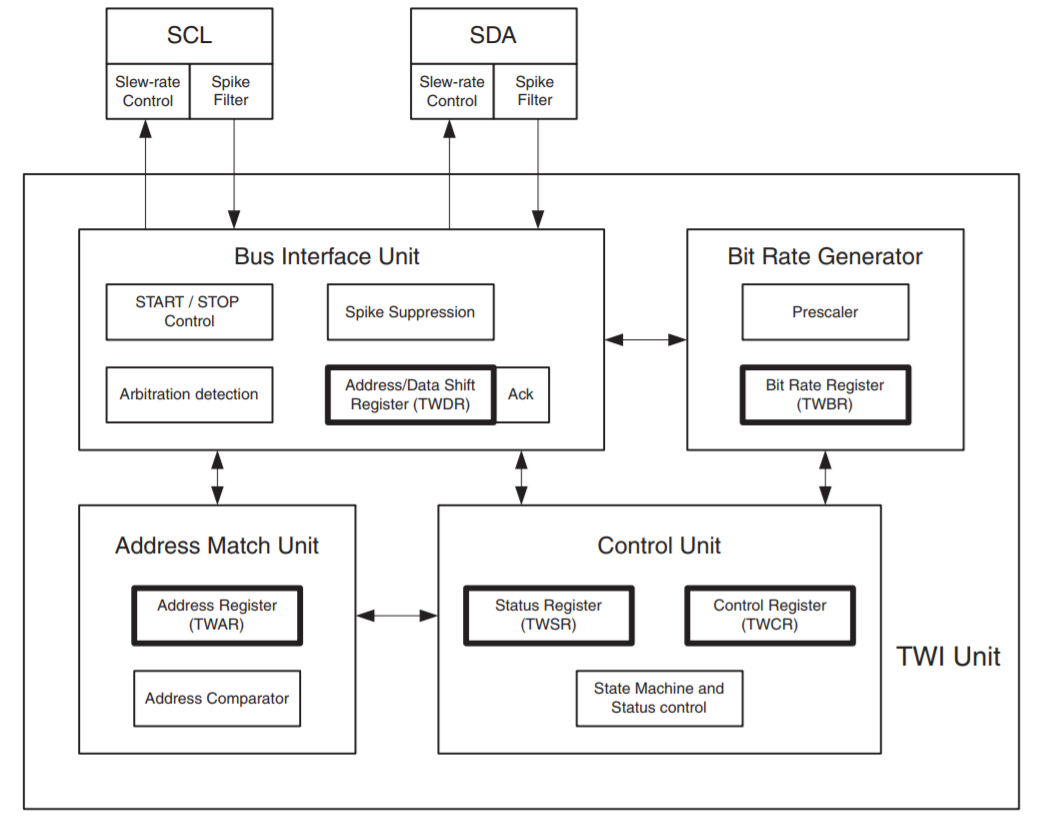
* Data Packet consists of 9 bits
  + First 8 bits indicates data to be transmitted/received
  + 9th bit is ACK (Acknowledge) bit
    - ACK = 1 🡪 Indicates Slave has not acknowledged transmission
    - ACK = 0 🡪 Indicates Slave has acknowledged transmission

Transmission:



* Transmission consists of a START condition, a SLA+R/W, one or more Data Packets and a STOP condition
* **A START followed by a STOP condition is illegal**

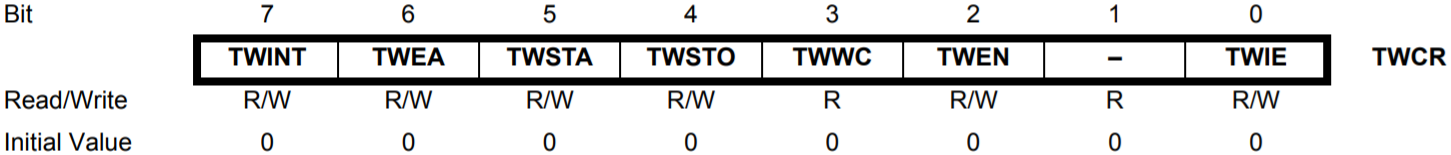
TWI Module:



* *SCL* (Serial Clock) [*PC0*] / *SDA* (Serial Data) [*PC1*] - Interfaces the TWI module with the rest of the MCU system
  + Slew-rate Control – Utilized for TWI specification conformity
  + Spike Filter – Suppresses spikes shorter than 50 ns
* Bit Rate Generator Unit – Controls *SCL* period when operating in Master Mode
  + Prescaler – Controlled by *TWPS* (Prescaler bits) in *TWSR* (Status Register)
  + Bit Rate Register (*TWBR*) – Controls *SCL* period by following formula:
* Bus Interface Unit – Interfaces with buses
  + Address/Data Shift Register (TWDR) – Address/Data byte to be sent or received. Contains an extra register for *(N)ACK* bit.
    - *(N)ACK* bit - Can be manipulated by *TWCR* (Control Register) when being received and by *TWSR* (Status Register) when being transmitted
  + START/STOP Control – Detects START/STOP conditions. If MCU is in sleep mode, the MCU is awaken and initiates Master Mode.
  + Arbitration detection – Continuously monitors transmission for false positives. Control unit is informed when an arbitration is lost
* Address Match Unit – Checks if received addresses match 7-bit address in *TWAR* (Address Register)
  + Address Register (*TWAR*) – Contains 7-bit address of MCU and the *TWGCE* (General Call Enable) bit
    - *TWGCE* bit – If set, will also compare addresses to general address
  + Address Comparator – Compares addresses
* Control Unit – Monitors the TWI bus and generates responses according to settings in *TWCR*
  + Control Register (*TWCR*) – Contains settings {explained later}
  + Status Register (*TWSR*) – Contains status flags {explained later}
    - When an event occurs requiring the attention of the application, the Interrupt Flag (*TWINT*) is asserted. A possible event is:
      * After the TWI has transmitted a START/REPEATED START
      * After the TWI has transmitted a SLA+R/W
      * After the TWI has transmitted an address byte
      * After the TWI has lost arbitration
      * After the TWI has been addressed by own address/general call
      * After the TWI has received a data byte
      * After a STOP/REPEATED START is received when slave
      * When an illegal START or STOP condition occurs

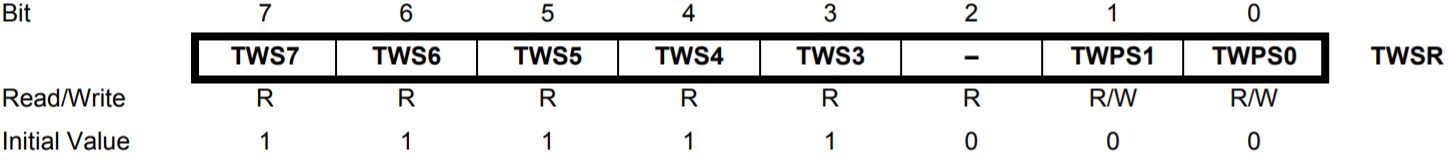
Register Description

*TWCR* – TWI Control Register



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bit # | Name | Description | Initial | R/W? |
| 7 | *TWINT* | **TWI Interrupt Flag**  Set by hardware when TWI has finished current job and expects application response   * If TWIE and the I-bit in SREG are both set, the MCU will jump to the *TWI* Interrupt Vector * While set, SCL remains low * Cleared by writing a logic one to it (NOT automatically when interrupt is completed) | 0 | R/W |
| 6 | *TWEA* | **TWI Enable Acknowledge Bit**  Controls generation of Acknowledge pulse  When written to one, the ACK pulse is generated:   * Own slave address is received * General call is received (*TWGCE* = 1 in *TWAR*) * A data byte is received   When written to zero, the device is virtually disconnected | 0 | R/W |
| 5 | *TWSTA* | **TWI START Condition Bit**  Controls START condition generation  When written to one, this device sends a START condition when the bus is idle. If not idle, the device waits until a STOP condition occurs, then claims the line.  **MUST BE CLEARED AFTER START CONDITION GENERATED** | 0 | R/W |
| 4 | *TWSTO* | **TWI STOP Condition Bit**  Controls STOP condition generation  When written to one, this device sends a STOP condition. This bit is cleared automatically when the STOP condition is generated. | 0 | R/W |
| 3 | *TWWC* | **TWI Write Collision Flag**  Signals a data write error  Set when attempting to write to TWDR when TWINT is low  Cleared when writing to TWDR when TWINT is high | 0 | R/W |
| 2 | *TWEN* | **TWI Enable Bit**  Controls TWI operation  When written to one, activates TWI interface, which takes control of *SCL* and *SDA* pins  When written to zero, terminates all TWI transmissions | 0 | R/W |
| 1 | *-* | Reserved (Always read at 0) | 0 | R |
| 0 | *TWIE* | **TWI Interrupt Enable**  Controls TWI Interrupt Capability  When written to one (I-bit = 1 in SREG), enables *TWI* interrupt | 0 | R/W |

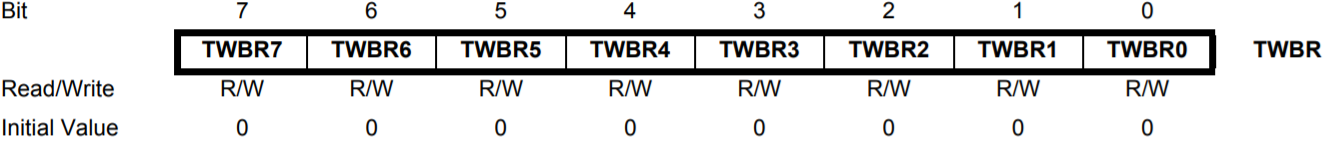
*TWSR* – TWI Status Register



*TWSR*[7:3] – Status Codes (see next page)

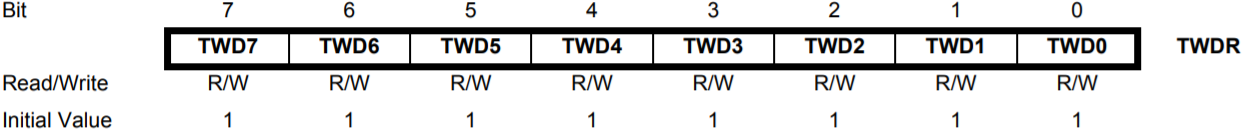
*TWSR*[1:0] – Prescaler for Bit Rate Generator (see next page)

TSBR – TWI Bit Rate Register



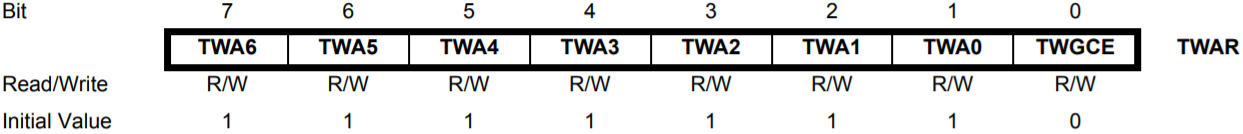
*TSBR*[7:0] – Bit Rate Division Factor Selection for Bit Rate Generator

*TWDR* – TWI Data Register



*TWDR*[7:0] – Next byte to be transmitted or last byte that was received. Writable only when TWINT is set by hardware.

*TWAR* – TWI (Slave) Address Register



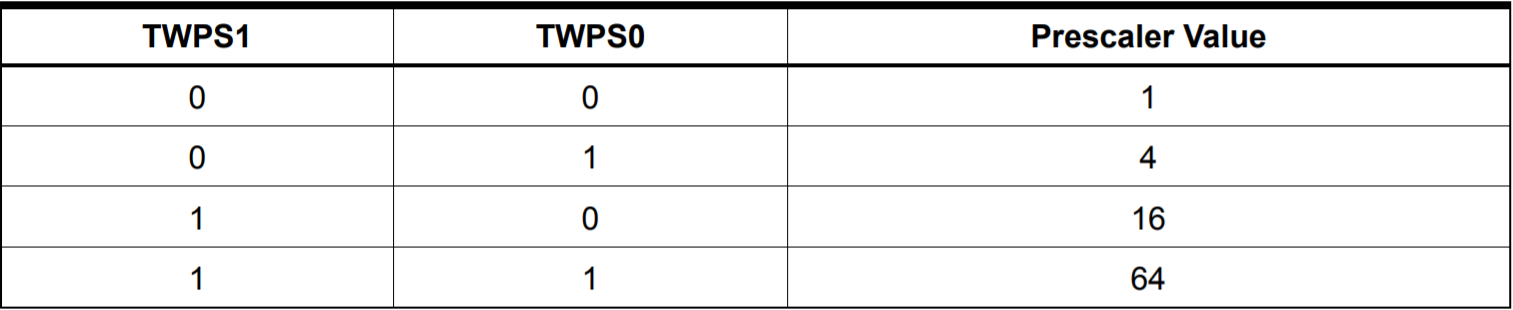
*TWAR*[7:1] – Slave address of device to be master of this device.

*TWAR*[0] – Enable recognition of general call address ($00)

* Generates interrupt for address match

Useful Resources

*TWPS* Bits (*TWSR*[1:0]) to Prescaler Values



Bit Rate Frequency Formula

Status Codes

Master Transmitter Mode:

<http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8154-8-bit-AVR-ATmega16A_Datasheet.pdf#page=178>

Master Receiver Mode

<http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8154-8-bit-AVR-ATmega16A_Datasheet.pdf#page=181>

Slave Receiver Mode

<http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8154-8-bit-AVR-ATmega16A_Datasheet.pdf#page=183>

Slave Transmitter Mode

<http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8154-8-bit-AVR-ATmega16A_Datasheet.pdf#page=186>

Typical Data Transmission Example

<http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8154-8-bit-AVR-ATmega16A_Datasheet.pdf#page=174>