GENERAL

assembler code examples i2c avr datasheet application

Published: September 19, 2026 | Index ID: #-

Understanding I2C on AVR with Assembly: A Comprehensive Guide

In the world of microcontrollers, the AVR family from Atmel (now Microchip) remains a favorite for hobbyists and professional engineers alike. One of the most widely used communication protocols in embedded systems is I2C (Inter Integrated Circuit). While many developers rely on high level libraries or C wrappers, there are compelling reasons to dive into assembler code: fine grained timing control, minimal footprint, and a deeper understanding of the hardware.

This article walks you through the essential concepts of I2C on AVR, how to read the datasheet, and how to craft efficient assembly routines. Whether you're building a sensor interface, a real time clock module, or a multi device bus, the examples and explanations below will give you the confidence to write clean, maintainable assembler code for your AVR projects.

1. The AVR I2C Interface at a Glance

AVR microcontrollers provide a dedicated hardware module called the Two Wire Interface (TWI) that implements the I2C protocol. The TWI module handles the low level details of start/stop conditions, address matching, and data transfer, while the firmware manages higher level logic such as error handling, retries, and buffer management.

- Master mode – The AVR initiates communication, sends addresses, and reads/writes data.
- Slave mode – The AVR responds to requests from a master, typically used for sensor data or configuration registers.
- Clock stretching – The slave can hold the SCL line low to delay the master until it's ready.

Because the TWI module is hardware based, the assembly code you write is responsible for configuring registers, handling interrupts, and orchestrating data flow.

2. Reading the AVR Datasheet: The Key to Mastery

Before writing any assembler code, you must become fluent in the AVR datasheet. The TWI section contains all the information you need: register definitions, bit fields, timing diagrams, and recommended usage patterns.

Here's how to approach the datasheet:

1. Locate the TWI section. It's usually found under "Peripheral description" or "Serial communication."
2. Identify the register map. Registers such as TWBR (bit rate), TWSR (status), TWCR (control), and TWDR (data) are central to your code.
3. Understand status codes. Each I2C operation yields a status code that indicates success or the type of error. The status register (TWSR) is split into prescaler bits and status bits.
4. Pay attention to timing tables. The datasheet provides minimum high/low times for SDA/SCL, which help you calculate the correct bit rate.
5. Review the application notes. Many AVR manufacturers publish application notes that illustrate typical use cases and best practices.

By mastering the datasheet, you'll avoid common pitfalls such as incorrect prescaler settings or misinterpreted

status codes.

3. Setting Up the TWI Module in Assembly

Below is a minimal initialization routine that prepares the AVR for I2C communication. It demonstrates register writes, bit manipulation, and the use of the ldi instruction.

```
; Initialize TWI (I2C) on AVR
; Assumes AVR with TWI module (ATmega328P, ATmega2560, etc.)

; Set bit rate: TWBR = 0x48 (for 100 kHz at 16 MHz)
ldi     r16, 0x48          ; Load 0x48 into r16
out     TWBR, r16          ; Write to TWBR register

; Set prescaler to 1: TWSR = 0x00
ldi     r16, 0x00
out     TWSR, r16

; Enable TWI, set master mode, enable ACK
ldi     r16, (1
```

In this snippet, TWEN enables the TWI module, TWIE enables the TWI interrupt (optional but useful for non blocking operation), and TWEA ensures the AVR will send ACKs automatically after each byte.

4. Master Transmit: Sending Data to a Slave

When the AVR acts as a master transmitter, the sequence of events is:

1. Generate a start condition.
2. Send the slave address with the write bit.
3. Transmit data bytes.
4. Generate a stop condition.

Below is a compact assembler routine that writes a single byte to a slave device. It checks status codes after each operation to ensure reliability.

```
; Transmit a single byte to a slave
; Parameters:
;   r18 = slave address (7  bit)
;   r19 = data byte to send

; 1. Start condition
ldi     r16, (1
```

This routine demonstrates the importance of checking the TWSR status after each step. It also shows how to use the TWSTO bit to generate a stop condition without waiting for an interrupt.

5. Master Receive: Reading Data from a Slave

Receiving data is slightly more involved because the master must acknowledge each byte except the last one. The following routine reads two bytes from a slave device and stores them in registers r20 and r21.

```
; Master receive two bytes from a slave
; Parameters:
;   r18 = slave address (7  bit)

; 1. Start condition
ldi     r16, (1
```

Notice the deliberate clearing of the TWEA bit before receiving the last byte. This instructs the slave that the master has finished reading, thereby sending a NACK to terminate the transfer.

6. Slave Example: Responding to Master Requests

In slave mode, the AVR must respond to address matches and data requests. The following routine demonstrates a simple slave that echoes any data it receives back to the master. It uses the TWI interrupt to keep the main loop free.

```
; TWI ISR for slave echo
; Assumes the ISR vector is defined elsewhere

; Check status
in      r16, TWSR
andi    r16, 0xF8

; 0x60: Own SLA+R received, ACK returned
cpi     r16, 0x60
breq    ack_sla_r
; 0x80: Data received, ACK returned
cpi     r16, 0x80
breq    data_received
; 0xA8: Own SLA+W received, ACK returned
cpi     r16, 0xA8
breq    ack_sla_w
rjmp    default_handler

ack_sla_r:
; Master requests data; load data into TWDR
```

Baca Juga (Artikel Terkait)

- **applied mechanics for engineering technology keith m walker**
  URL: http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-keith-m-walker.pdf
- **applied mechanics for engineering technology answers**
  URL: http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-answers.pdf
- **applied mechanics for engineering technology 8th edition solution manual**
  URL: http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-8th-edition-solution-manual.pdf
- **applied mechanics for engineering technology 8th edition**
  URL: http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-8th-edition.pdf

Tags: #assembler #code #examples #datasheet #application