2. Inter-Integrated Circuit (I2C)

2.1 Overview

I2C is a synchronous two-wire serial bus:

- SDA: Serial Data
- SCL: Serial Clock

I2C supports multiple devices on the same bus using slave addressing and ACK/NACK handshaking.

2.2 I2C Transaction Flow

1. Master sends START condition.
2. Master sends Slave Address + R/W bit.
3. Slave replies with ACK.
4. Data bytes are exchanged with ACK/NACK after each byte.
5. Master sends STOP condition.

2.3 I2C Speed Modes

- Standard mode: up to 100 kHz
- Fast mode: up to 400 kHz
- Fast mode plus: up to 1 MHz
- High-speed mode: up to 3.4 MHz
- Ultra-fast mode (unidirectional): up to 5 MHz

2.4 I2C Registers (Detailed Bits)

TWSR - TWI Status Register

Function:

- Contains status code bits (TWS7:TWS3) and prescaler bits.

Bit	Name	Description
7	TWS7	Status code bit 7
6	TWS6	Status code bit 6

Bit	Name	Description
5	TWS5	Status code bit 5
4	TWS4	Status code bit 4
3	TWS3	Status code bit 3
2	Reserved	Reserved/unused
1	TWPS1	Prescaler select bit 1
0	TWPS0	Prescaler select bit 0

Prescaler mapping:

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

TWBR - TWI Bit Rate Register

Function:

- Sets the SCL clock generator division factor in master mode.

Bit	Name	Description
7	TWBR7	Bit-rate divider bit 7
6	TWBR6	Bit-rate divider bit 6
5	TWBR5	Bit-rate divider bit 5
4	TWBR4	Bit-rate divider bit 4
3	TWBR3	Bit-rate divider bit 3
2	TWBR2	Bit-rate divider bit 2
1	TWBR1	Bit-rate divider bit 1
0	TWBR0	Bit-rate divider bit 0

Clock formula:

SCL = F_CPU / (16 + 2 x TWBR x PrescalerValue)

TWCR - TWI Control Register

Function:

- Controls start/stop generation, ACK, interrupt, and enable state.

Bit	Name	Description
7	TWINT	TWI interrupt flag (set by hardware when operation completes)
6	TWEA	TWI enable acknowledge
5	TWSTA	TWI start condition request
4	TWSTO	TWI stop condition request
3	TWWC	TWI write collision flag
2	TWEN	TWI enable
1	Reserved	Reserved
0	TWIE	TWI interrupt enable

TWDR - TWI Data Register

Function:

- Holds the current transmitted/received byte.

Bit	Name	Description
7	TWD7	Data bit 7
6	TWD6	Data bit 6
5	TWD5	Data bit 5
4	TWD4	Data bit 4
3	TWD3	Data bit 3
2	TWD2	Data bit 2
1	TWD1	Data bit 1
0	TWD0	Data bit 0

TWAR - TWI Address Register

Function:

- Holds own slave address and general call control.

Bit	Name	Description
7	TWA6	Slave address bit 6
6	TWA5	Slave address bit 5

Bit	Name	Description
5	TWA4	Slave address bit 4
4	TWA3	Slave address bit 3
3	TWA2	Slave address bit 2
2	TWA1	Slave address bit 1
1	TWA0	Slave address bit 0
0	TWGCE	General call recognition enable

Common TWI Status Codes

Status Code	Meaning
0x08	START condition transmitted
0x10	Repeated START transmitted
0x18	SLA+W transmitted, ACK received
0x20	SLA+W transmitted, NACK received
0x28	Data transmitted, ACK received
0x30	Data transmitted, NACK received
0x38	Arbitration lost
0x40	SLA+R transmitted, ACK received
0x48	SLA+R transmitted, NACK received
0x50	Data received, ACK returned
0x58	Data received, NACK returned

2.5 I2C Assembly Examples

I2C Master Transmit

```
;-------------------------
; Assembly Code - Master Tx
;-------------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;-----------------------
.global main
;============================================================
main:
```

```
    CBI   DDRC, 3         ;pin PC3 is i/p
    ;-------------------------------------------------------
    RCALL I2C_init        ;initialize TWI module
    ;-------------------------------------------------------
l1: SBIS  PINC, 3
    RJMP  l1              ;wait for "transmit" button press
    ;-------------------------------------------------------
    RCALL I2C_start       ;transmit START condition
    LDI   R27, 0b10010000 ;SLA(1001000) + W(0)
    RCALL I2C_write       ;write slave address SLA+W
    LDI   R27, 0b11110101 ;data byte to be transmitted
    RCALL I2C_write       ;write data byte
    RCALL I2C_stop        ;transmit STOP condition
    ;-------------------------------------------------------
    RJMP  l1              ;go back for another transmit
;=============================================================
I2C_init:
    LDI   R21, 0
    STS   TWSR, R21       ;prescaler = 0
    LDI   R21, 12         ;division factor = 12
    STS   TWBR, R21       ;SCK freq = 400kHz
    LDI   R21, (1<<TWEN)
    STS   TWCR, R21       ;enable TWI
    RET
;=============================================================
I2C_start:
    LDI   R21, (1<<TWINT)|(1<<TWSTA)|(1<<TWEN)
    STS   TWCR, R21       ;transmit START condition
    ;-------------------------------------------------------
wt1:LDS   R21, TWCR
    SBRS  R21, TWINT      ;TWI interrupt = 1?
    RJMP  wt1             ;no, wait for end of transmission
    ;-------------------------------------------------------
    RET
;=============================================================
I2C_write:
    STS   TWDR, R27       ;copy SLA+W into data register
    LDI   R21, (1<<TWINT)|(1<<TWEN)
    STS   TWCR, R21       ;transmit SLA+W
    ;-------------------------------------------------------
```

```
wt2:LDS   R21, TWCR
    SBRS  R21, TWINT
    RJMP  wt2             ;wait for end of transmission
    ;-------------------------------------------------------
    RET
;=============================================================
I2C_stop:
    LDI   R21, (1<<TWINT)|(1<<TWSTO)|(1<<TWEN)
    STS   TWCR, R21       ;transmit STOP condition
    RET
```

I2C Slave Receive

```
;-------------------------
; Assembly Code - Slave Rx
;-------------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;------------------------
.global main
;=============================================================
main:
    LDI   R21, 0xFF
    OUT   DDRD, R21         ;port D is o/p
    CBI   DDRC, 3           ;pin PC3 is i/p
;-------------------------------------------------------------
agn:RCALL I2C_init          ;initialize TWI module
    RCALL I2C_listen        ;listen to bus to be addressed
    RCALL I2C_read          ;read data byte
    OUT   PORTD, R27        ;and o/p to port D
;-------------------------------------------------------------
l1: SBIS  PINC, 3
    RJMP  l1                ;wait for "listen" button press
;-------------------------------------------------------------
    LDI   R26, 0
    OUT   PORTD, R26        ;clear port D
    RJMP  agn               ;& go back & listen to bus
;=============================================================
I2C_init:
    LDI   R21, 0b10010000
    STS   TWAR, R21         ;store slave address 0b10010000
```

```
    LDI   R21, (1<<TWEN)
    STS   TWCR, R21         ;enable TWI
    LDI   R21, (1<<TWINT)|(1<<TWEN)|(1<<TWEA)
    STS   TWCR, R21         ;enable TWI & ACK
    RET
;=================================================================
I2C_listen:
    LDS   R21, TWCR
    SBRS  R21, TWINT
    RJMP  I2C_listen        ;wait for slave to be addressed
    RET
;=================================================================
I2C_read:
    LDI   R21, (1<<TWINT)|(1<<TWEA)|(1<<TWEN)
    STS   TWCR, R21         ;enable TWI & ACK
    ;-----------------------------------------------------------
wt: LDS   R21, TWCR
    SBRS  R21, TWINT
    RJMP  wt                ;wait for data byte to be read
    ;-----------------------------------------------------------
    LDS   R27, TWDR         ;store received byte
    RET
```

Revision #1
Created 2026-04-18 15:10:18 UTC by AX
Updated 2026-04-18 15:10:59 UTC by AX