Mechatronics Engineering

I2C Communication Protocol

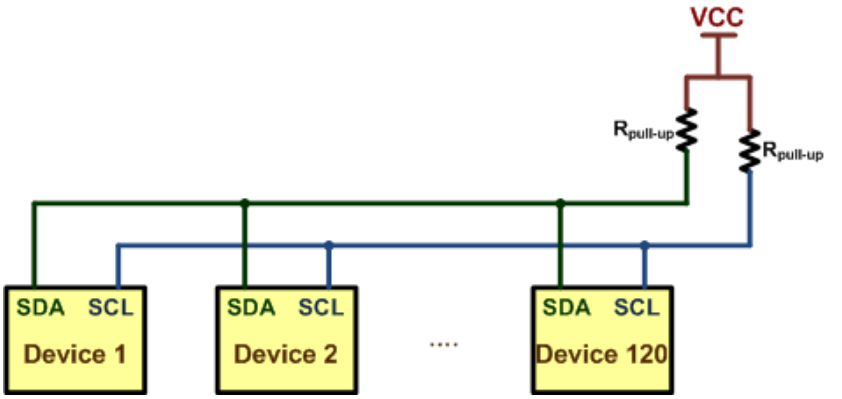
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Materials Science
German University in Cairo

I2C History

- IIC: Inter-Integrated Circuit
- Philips
- 1982
- Aim: connecting many devices (around 128 devices) to the MCU using two wires

Connecting devices using I2C

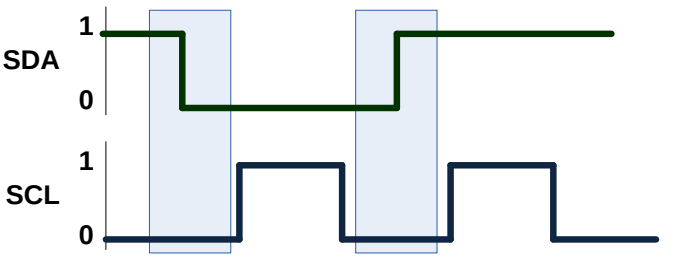
- SDA: Serial Data
- SCL: Serial Clock
- Open-collector devices (Wire AND)



GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Sending bits of data

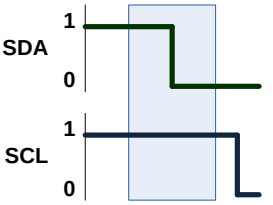
- The SDA values changes when SCL is low.
- The receiver reads SDA on the falling edge of SCL.



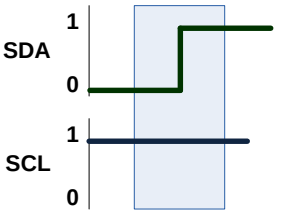
GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Start and Stop conditions

- Start



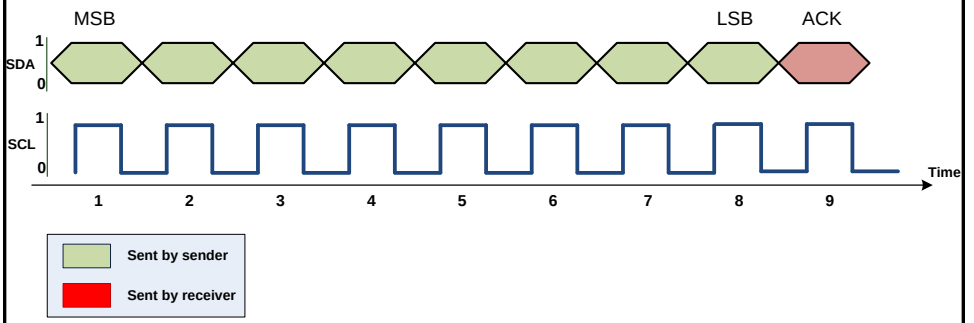
- Stop



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Packet Format

- Each packet is 9 bits long.
- First 8 bits are put on SDA by the transmitter
- The 9th bit is an acknowledge by the receiver
  - NACK (leave high) or ACK (pull down)



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

Master vs. Slave

- Master
  - Begins the communication
  - Chooses the slave
  - Makes clock
  - Sends or receives data
- Slave
  - Responds to the master
  - Each slave has a unique 7-bit address

Master vs. Slave (Cont.)

- There might be more than 1 master on an I2C bus
- Each device can be both Master and Slave

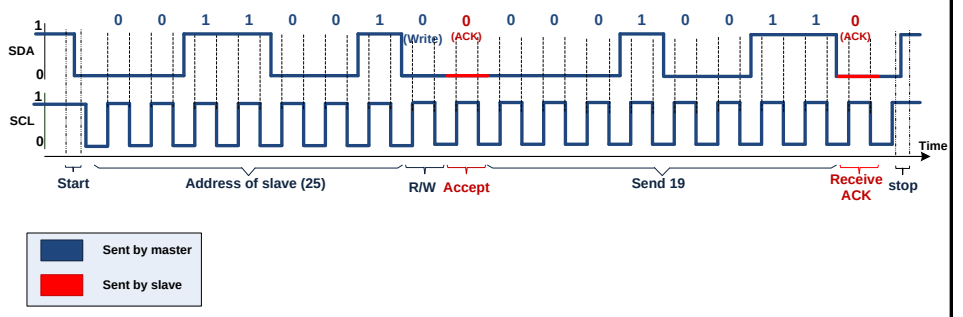
Steps of a communication

1. Start
2. Address
3. Send or Receive (Write or read)
4. Acknowledge
5. Send/receive a byte of data
6. Acknowledge
7. Stop

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

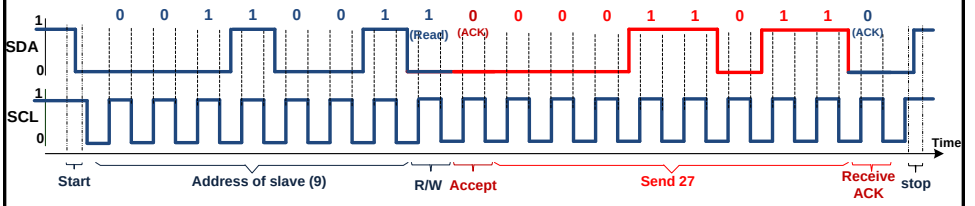
Sending a byte

- Sending 19 to device 25.



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

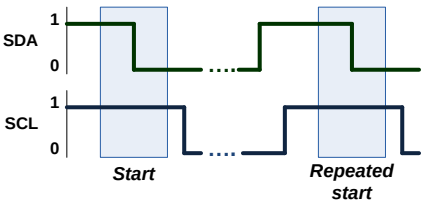
Receiving a byte from Device 9



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

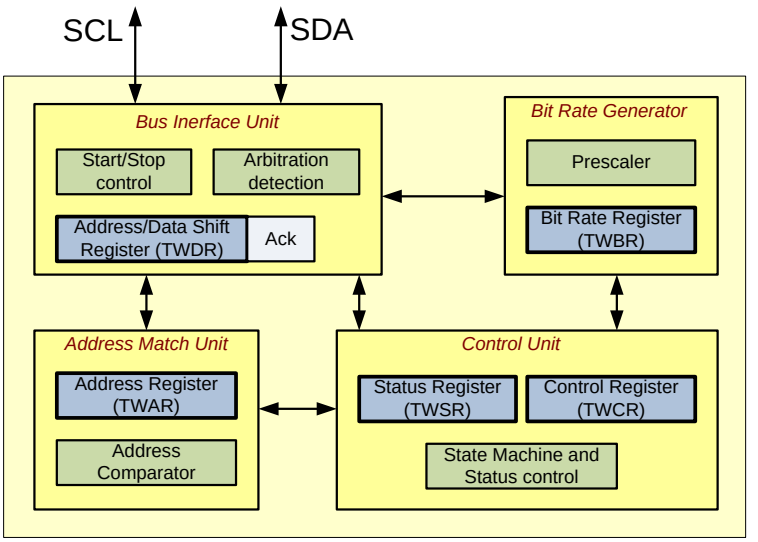
Repeated start

- A new Start condition before the Stop condition



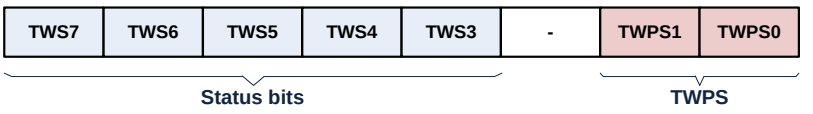
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

I2C (TWI) Unit in AVR



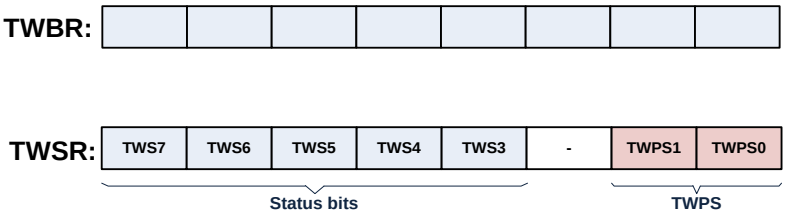
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

TWSR (TWI Status Register)



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

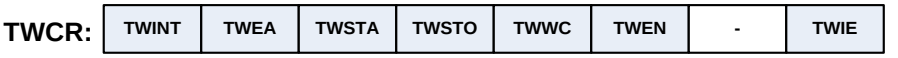
TWBR (TWI Bit Rate) Register



$$frequency = \frac{XTAL}{16 + 2 \times TWBR \times 4^{TWPS}}$$

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

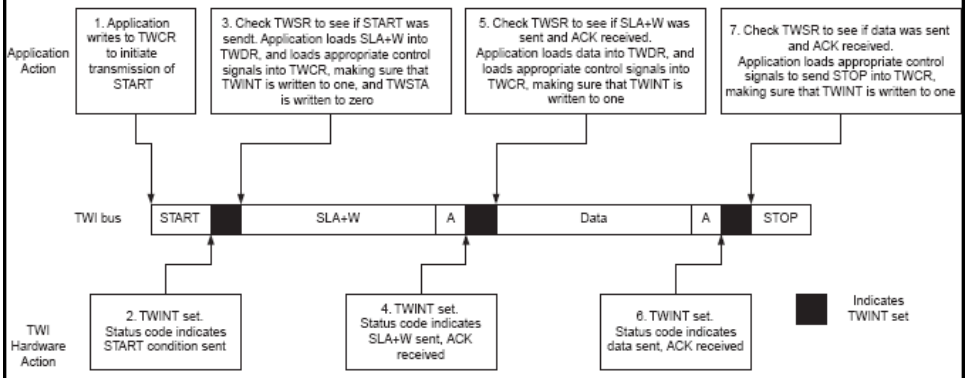
TWCR



- TWINT: TWI Interrupt flag
- TWEA: TWI Enable Acknowledge bit
  - 1:ACK, 0:NACK
- TWSTA: TWI Start condition bit
- TWSTO: TWI Stop condition bit
- TWWC: TWI Write Collision flag
- TWEN: TWI Enable bit
- TWIE: TWI Interrupt Enable

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

TWI Control Register



Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

TWAR (TWI Address Register)

TWAR:

TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE

- TWA6-0 (TWI slave Address)
- TWGCE (TWI General Call Recognition Enable bit)
  - 1: Answer to general call

Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

TWDR (TWI Data Register)

TWI, Master Mode programming

- Initializing
  - Set the TWI module clock frequency by setting the values of the TWBR register and the TWPS bits in the TWSR register.
  - Set the TWEN bit in TWCR to one to enable the TWI module
- Transmit START condition
  - Set TWEN, TWSTA, and TWINT bits of TWCR to one.

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science

TWI, Master Mode programming

- Send Data
  - Copy the data byte to the TWDR
  - Set the TWEN and TWINT bits of the TWCR to one to start sending the byte.
  - Poll TWINT flag in TWCR register to see whether the byte transmitted completely
- Receive Data
  - Set TWEN and TWINT bits of TWCR to one to start receiving a byte.
  - Poll TWINT flag in TWCR to see whether a byte has been received completely
  - read the received byte from the TWDR

TWI, Master Mode programming

- Transmit STOP condition
  - Set TWEN, TWSTO, and TWINT bits of TWCR to one
    - Note: we cannot poll the TWINT flag after transmitting the STOP condition

Writing and reading a byte in master mode

```
#include <avr/io.h>

void i2c_write(unsigned char data)
{
  TWDR = data ;
  TWCR = (1<< TWINT)|(1<<TWEN);
  while ((TWCR & (1 <<TWINT)) == 0);
}

//*************************************************
unsigned char i2c_read(unsigned char isLast)
{
  if (isLast == 0)  //send ACK
    TWCR = (1<< TWINT)|(1<<TWEN)|(1<<TWEA); //send ACK
  else
    TWCR = (1<< TWINT)|(1<<TWEN); //send NACK
  while ((TWCR & (1 <<TWINT)) == 0);
  return TWDR;
}
```



```
void i2c_start(void)
{
  TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN);
  while ((TWCR & (1 << TWINT)) == 0);
}

void i2c_stop()
{
  TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWSTO);
}

void i2c_init(void)
{
  TWSR=0x00; //set prescaler bits  to zero
  TWBR=152;  //SCL frequency is 50K for XTAL = 16M
  TWCR=0x04; //enable the TWI module
}
```

```
int main (void)
{
  i2c_init();

  //writing a byte
  i2c_start();              //transmit START condition
  i2c_write(0b11010000);    //transmit SLA + W(0)
  i2c_write(0b11110000);    //transmit data
  i2c_stop();               //transmit STOP condition

  //reading a byte
  i2c_start();              //transmit START condition
  i2c_write(0b11010001);    //transmit SLA + R(1)
  PORTD =i2c_read(1);       //read one byte of data
  i2c_stop();               //transmit STOP condition
  while(1);                 //stay here forever
}
```


TWI, Slave Mode programming

- Initializing
  - Set the slave address by setting the values for the TWAR register.
    - 7 bits for address
    - 8th bit is TWGCE (1 = answer general calls)
  - Set the TWEN bit in TWCR to one to enable the TWI module
  - Set the TWEN, TWINT, and TWEA bits of TWCR to one to enable the TWI and acknowledge generation
- Listening
  - poll the TWINT flag to see when the slave is addressed by a master device or use its interrupt

TWI, Slave Mode programming

- Send Data
  - Copy the data byte to the TWDR
  - Set the TWEN, TWEA, and TWINT bits of the TWCR register to one to start sending the byte.
  - Poll TWINT flag in TWCR register to see whether the byte transmitted completely
- Receive Data
  - Set TWEN and TWINT bits of TWCR to one to start receiving a byte.
  - Poll TWINT flag in TWCR to see whether a byte has been received completely
  - read the received byte from the TWDR

GUC German University in Cairo
Prof. Ayman A. El-Badawy
Department of Mechatronics Engineering
Faculty of Engineering and Material Science