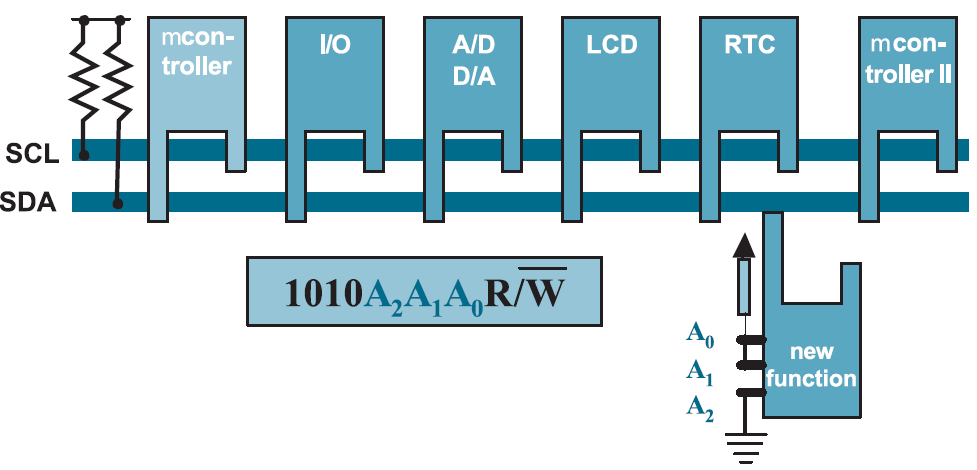
***Interfacing I2C EEPROM***

***Description:***

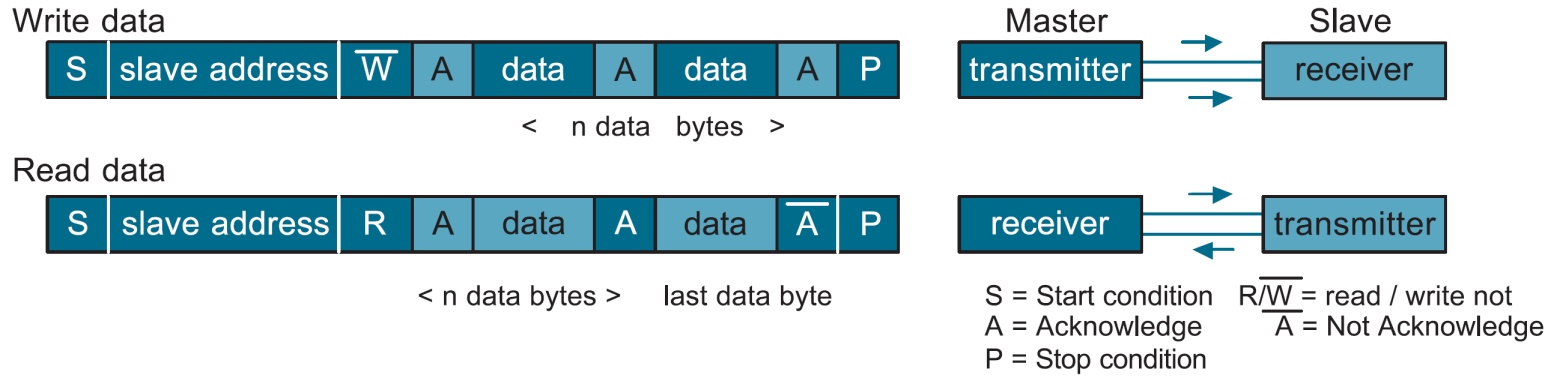
*I²C is an abbreviation of Inter Integrated Circuit and is a protocol for serial communication between Integrated Circuits, it is also called Two Wire Interface (TWI). The bus is used for communication between microcontrollers and peripheral devices like memories, temperature sensors and I/O expanders. An EEPROM is a Electrically Erasable and Programmable Read Only Memory.*

|  |  |  |  |
| --- | --- | --- | --- |
| *EEPROM Model* | *Size* | *Internally Organized* | *Address (hex)* |
| AT24C01 | 128 Bytes | 128 x 8 = 1024 bits | *00000 >> 0007F* |
| AT24C02 | 256 Bytes | 256 x 8 = 2048 bits | *00000 >> 000FF* |
| AT24C04 | 512 Bytes | 512 x 8 = 4096 bits | *00000 >> 001FF* |
| AT24C08 | 1 Kbyte | 1024 x 8 = 8192 bits | *00000 >> 003FF* |
| AT24C16 | 2 Kbyte | 2048 x 8 = 16384 bits | *00000 >> 007FF* |
| AT24C32 | 4 Kbyte | 4096 x 8 = 32768 bits | *00000 >> 00FFF* |
| AT24C64 | 8 Kbyte | 8192 x 8 = 65536 bits | *00000 >> 01FFF* |
| AT24C128 | 16 Kbyte | 16384 x 8 = 131072 bits | *00000 >> 03FFF* |
| AT24C256 | 32 Kbyte | 32768 x 8 = 262144 bits | *00000 >> 07FFF* |
| AT24C512 | 64 Kbyte | 65536 x 8 = 524288 bits | *00000 >> 0FFFF* |
| AT24C1024 | 128 Kbyte | 131072 x 8 = 1048576 bits | *00000 >> 1FFFF* |

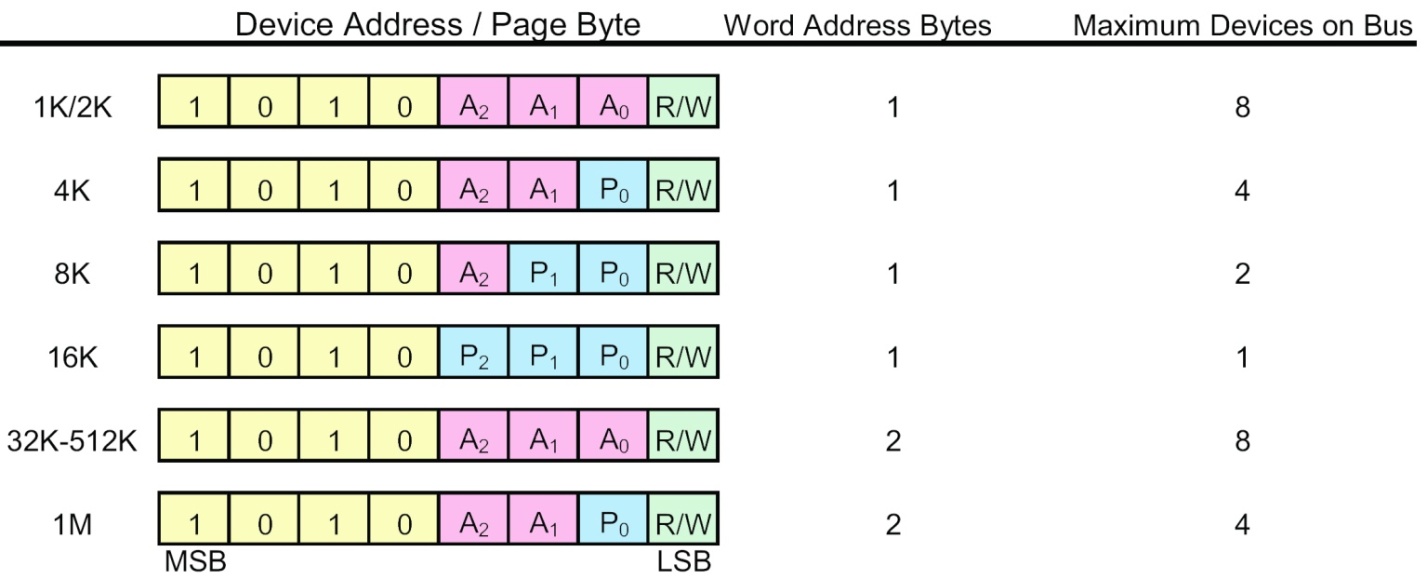
*The communication of the bus goes along two lines:* ***SDA*** *(Serial Data) and* ***SCL*** *(Serial Clock). Each I²C device has a unique* ***7-bit address*** *(Device Select Code). The most significant bits are fixed and assigned to a specific device category (e.g. b1010 is assigned to serial EEPROMS). The three less significant bits (****A2,A1 and A0****) are programmable and used to address the device. The three bits allows eight different I2C address combinations and therefore allowing up to eight different devices of that type to operate on the same I2C-bus. The I2C address is send in the 1st byte, the lest significant bit of the first byte is used to indicate if the master is going to* ***write(0) or read(1)*** *from the slave.*



*The device that sends data along the bus is called* ***master****, a device that receives the data is called* ***slave****. The master starts the transmission with a start signal and stops the transmission with a stop signal on the SDA line. During the start and stop signals the SCL line has to be high. After the master has started the data-transmission with a start signal, the master writes a device address byte to the slave. Each data byte has to have a length of 8 bits. The slave has to acknowledge the reception of the data byte with a acknowledge-bit (ACK).*

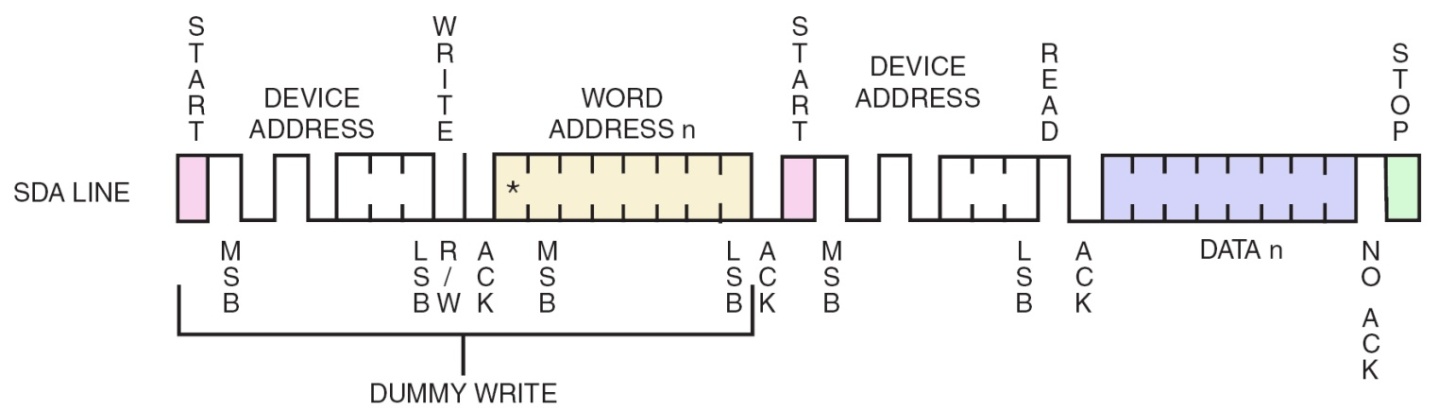


*A write operation requires a device address bytes, two address bytes and the data-byte. Upon receive of the address the EEPROM sends an ACK and then clocks in the data-byte. The EEPROM sends again an ACK and the microcontrollers sends a stop-signal to terminate the write sequence.*

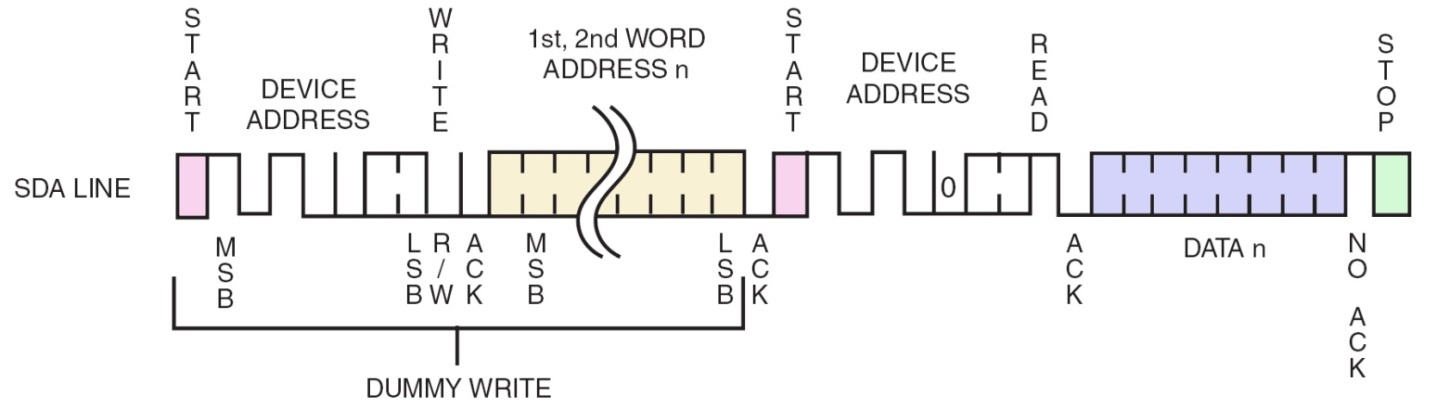


*All devices from 32K – 512K will require no system changes and can be interchanged with only the page size differences to consider.*

***Low Density Random Read:***

******

***Medium and High Density Random Read:***



***Software:***

*The BASCOM-AVR compiler is used to make a program that writes and reads one byte from the EEPROM. BASCOM has several embedded commands to control the I2C bus.*

*In BASCOM-AVR you first have to configure the ports you use for the SDA and SCL lines of the I2C bus. Then you send the device address to select the EEPROM that is connected to the I2C bus. After that you send two bytes to the EEPROM to select the address in the EEPROM to which you want to write the data. The last byte to send in a write sequence is the data byte.*

**$regfile** = "m16def.dat"  
**$crystal** = 2000000  
**$lib** "I2C\_TWI.LBX"  
**$baud** = 9600  
'---------------------------  
**Config** Scl = Portc.0  
**Config** Sda = Portc.1  
**Config** Twi = 100000 '100KHZ  
'---------------------------  
**Const** Addressw = 160 '&B10100000 slave write address  
**Const** Addressr = 161 '&B10100001 slave read address  
'---------------------------  
**Dim** Adres\_h **As** **Byte** , Adres\_l **As** **Byte**  
**Dim** Rd\_value **As** **Byte** , Wr\_value **As** **Byte**  
'---------------------------  
**Do**  
 **Input** "Wr\_value:" , Wr\_value  
 **Input** "Adres\_l:" , Adres\_l  
 **Input** "Adres\_h:" , Adres\_h  
  
 **Gosub** Write\_eeprom  
 **Gosub** Read\_eeprom  
  
 **Print** "Error W: " ; **Err**  
 **print** "Wr\_value: " ; Wr\_value  
  
 **Print** "Error R: " ; **Err**  
 **Print** "Rd\_value: " ; Rd\_value  
**Loop**  
**End**  
'---------------------------  
  
Write\_eeprom:  
 **I2cstart** 'Start condition  
 **I2cwbyte** Addressw 'Slave address  
 **I2cwbyte** Adres\_h 'H address of EEPROM   
 **I2cwbyte** Adres\_l 'L address of EEPROM  
 **I2cwbyte** Wr\_value 'Value to write  
 **I2cstop** 'Stop condition  
 **Waitms** 10 'Wait for 10 milliseconds  
**Return**  
'---------------------------  
Read\_eeprom:  
 **I2cstart** 'Generate start  
 **I2cwbyte** Addressw 'Slave adsress  
 **I2cwbyte** Adres\_h 'H address of EEPROM   
 **I2cwbyte** Adres\_l 'L address of EEPROM  
 **I2cstart** 'Repeated start  
 **I2cwbyte** Addressr 'Slave address (read)  
 **I2crbyte** Rd\_value , **Nack** 'Read byte  
 **I2cstop** 'Generate stop  
**Return**  
'---------------------------

AT24C32 (4 Kbyte)

***4096 \* 8 = 32768 bits 0000 >> 0FFF 32 byte page***

|  |  |  |
| --- | --- | --- |
| **&H0000** | Saturday [128 Set]  128 x 4 = 215 Bytes  512Bytes | **&H01FF** |
| **&H0200** | Sunday [128 Set]  128 x 4 = 215 Bytes  512Bytes | **&H03FF** |
| **&H0400** | Monday [128 Set]  128 x 4 = 215 Bytes  512Bytes | **&H05FF** |
| **&H0600** | Tuesday [128 Set]  128 x 4 = 215 Bytes  512Bytes | **&H07FF** |
| **&H0800** | Wednesday [128 Set]  128 x 4 = 215 Bytes  512Bytes | **&H09FF** |
| **&H0A00** | Thursday [128 Set]  128 x 4 = 215 Bytes  512Bytes | **&H0BFF** |
| **&H0C00** | Friday [128 Set]  128 x 4 = 215 Bytes  512Bytes | **&H0DFF** |
| **&H0E00** | NON USED AREA  512Bytes | **&H0FFF** |

[