SIEMENS

Microcontrollers

ApNote   AP1623

Software module of M3L-bus by using USART for C16x microcontroller family.

This is a software module of M3L-bus by using Asynchronous/Synchronous serial channel (USART) in synchronous mode for C16X microcontroller family. The M3L-bus is used mainly to communicate between the microcontroller and Television application IC like MegaText, and Compact Text.

Author: Tan Choon Hock/SCPL HL RM LAB

1 Introduction to M3L-bus

The M3L-bus is a single master protocol which consists of 3-wire bus; the SCL(clock), SDA(data), and I2CEN(for START and STOP condition). The clock frequency can achieve up to maximum 1.0Mhz that mean faster data transfer rate. The microcontrollers from Siemens have a dedicated hardware interface specially designed for M3L-bus protocol. This software module is using the dedicated hardware peripheral of Asynchronous/Synchronous Serial Interface(USART) of the C16X microcontroller family.

2 M3L-bus Specification

2.1 Data Transfer formats

When there is a HIGH to LOW transition on the I2CEN line, it indicates a START condition. A LOW to HIGH transition on the I2CEN is defined as the STOP condition. The data line can only be changed when the clock signal on the SCL line is HIGH. Therefore, the data on the SDA line must be stable during the LOW period of the clock signal.

Each information puts on the SDA line must be 8-bit long. There is no acknowledge bit followed by every byte sent for this M3L-bus protocol. The reason for not implement the acknowledge bit is that the MegaText IC comes with input filter in the SDA and SCL lines which will suppress high-frequency interference on those lines.

In M3L-bus mode, the LSB of the data is transmitted first. For example, if the chip selected address is MSB R/W 1000100 LSB, then the data on the SDA line will be LSB 0010001(R/W) MSB. The difference from I2C is that the READ/WRITE bit is on the MSB position. The write select address will be 44H, and read select address in this case is C4H. If the transmitted or received data is MSB 10110001 LSB, the data on the SDA line will be LSB 10001101 MSB. The USART peripheral of the Siemens' microcontroller will be sending and receiving the LSB bit first. The data of transmit and receive shift registers are in the normal format MSB xxxxxxxx LSB. Therefore, there is no additional software involved for any conversion of data.

There are only two basic types of data transfer format for M3L-bus. Figure #1 shows the M3L -bus data transfer format of writing data from microcontroller/master to MegaText/slave. Figure #2 shows the data transfer format of reading data from the MegaText/slave to microcontroller/master.
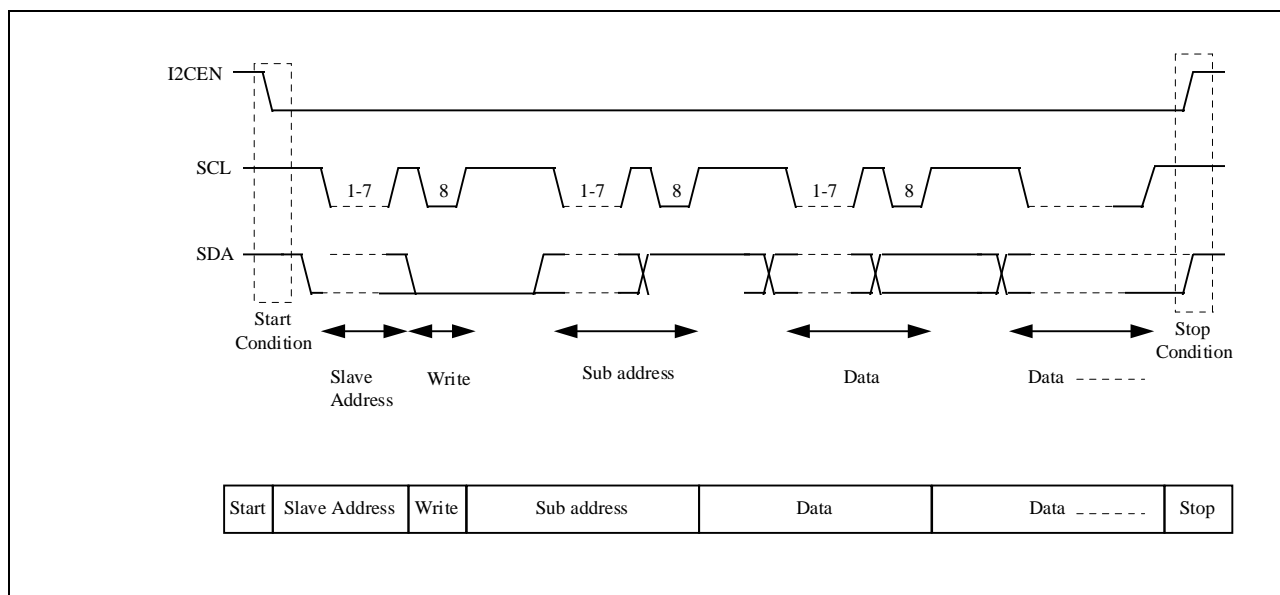
Figure 1:

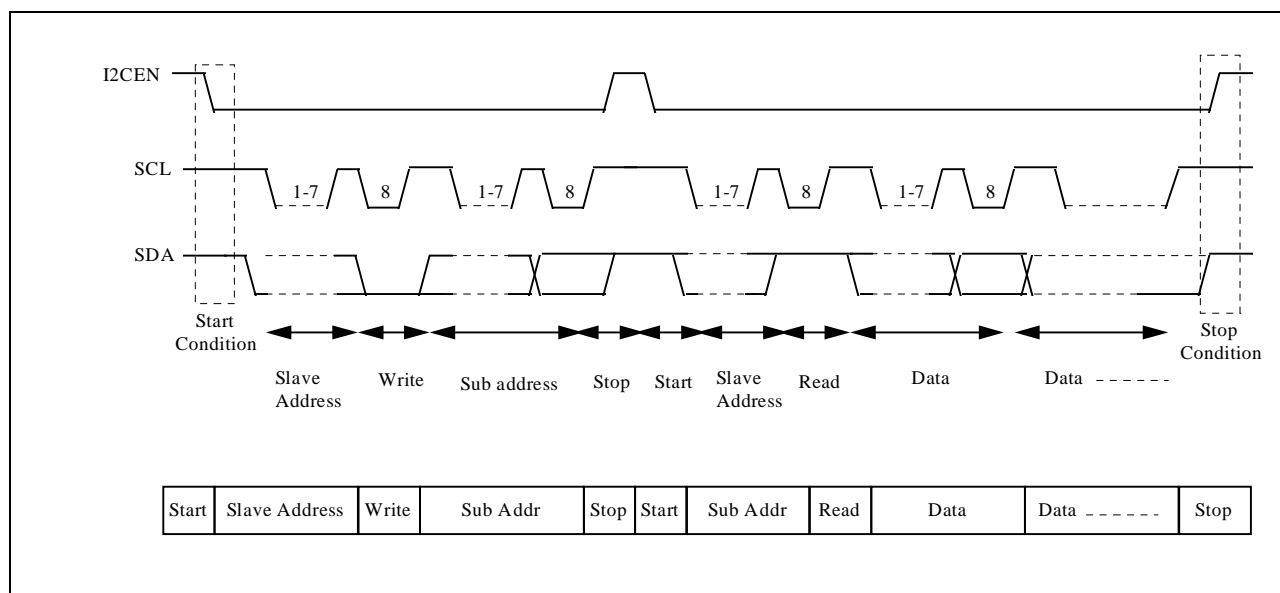**M3L-bus data transfer format of writing data to slave**

Figure 2:

M3L-bus data transfer format of reading data from slave

SIEMENS

2.2 Timing Diagram

The clock frequency of SCL is in the range of 0 up to 1.0Mhz. The clock on the M3L-bus has a minimum LOW and HIGH period of 400ns.

Occasionally, the MegaText/slave device may slow down the transmission by holding the clock line low after receiving a byte of data from microcontroller. This phenomenon is defined as a WAIT condition. Therefore, microcontroller/master needs to switch the SCL output to high impedance and read the SCL line before transmitting another byte of data to the slave device.

Figure #3 shows the data transfer timing requirements in detail. The description of the abbreviations used is shown in the Table #1. The minimum timing requirements are needed to be fulfilled in order for M3L-bus to operate properly.
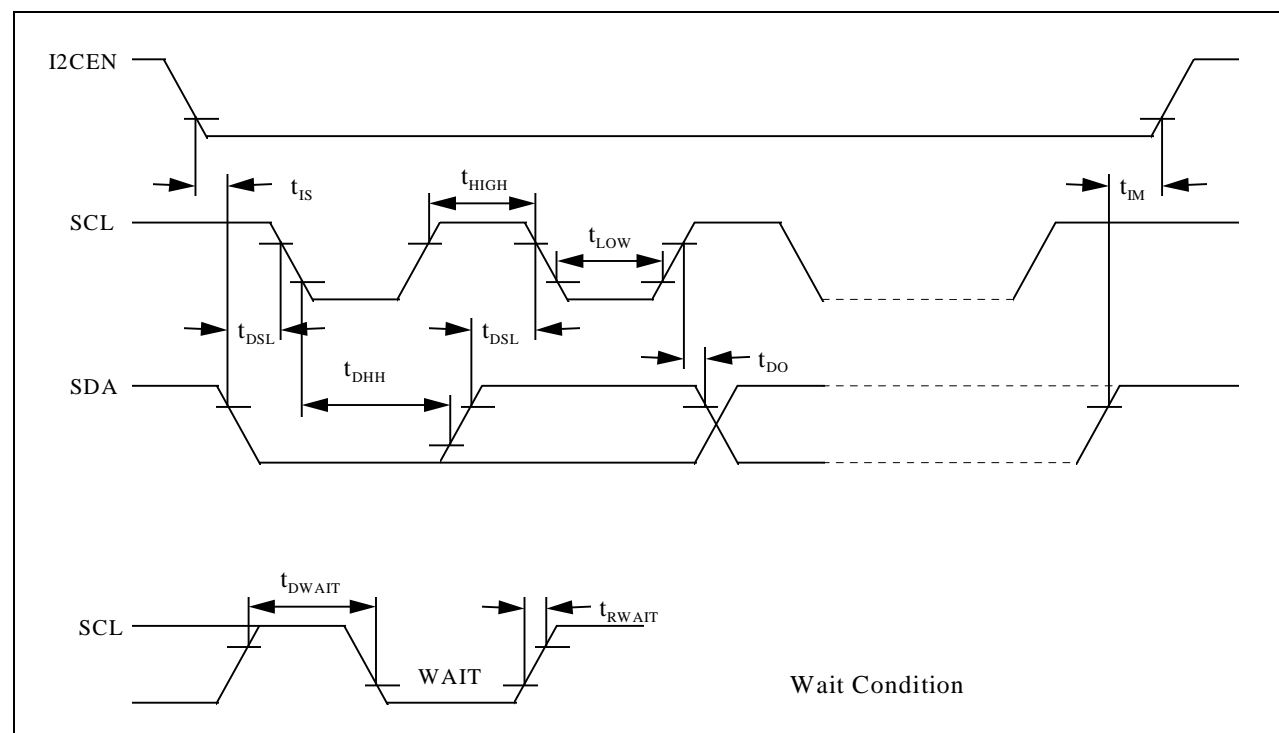

Figure 3:

M3L-bus timing diagram

Table 1:

Abbreviation for M3L-bus timing diagram

Parameter	Symbol	Limit Values			Unit
		min.	typ.	max.	
1. Set-up time I2CEN to SDA falling edge	t_{IS}	400			ns
2. Set-up time SDA input to SCL falling edge	t_{DSL}	100			ns
3. Hold time for SDA input from SCL falling edge to the next rising edge of SCL	t_{DHH}	400			ns
4. HIGH-time clock	t_{HIGH}	400			ns
5. LOW-time clock	t_{LOW}	400			ns
6. Delay from SCL rising edge until SDA open drain output stage changes impedance	t_{DO}	250		500	ns
7. Set-up time from SDA to I2CEN rising edge	t_{IM}	400			
8. Delay from SCL rising edge to SCL forced low for WAIT condition	t_{DWAIT}	500		750	ns
9. SCL pull-up time at the end of WAIT condition	t_{RWAIT}	70		100	ns
10. SCL load capacitance	C_{SCL}			200	pF
11. SDA low level output impedance			100		ohm

SIEMENS

2.3   Hardware Connection

Every device connected to the M3L-bus must have an open drain/open collector output for both the clock(SCL) and data(SDA) lines. Each of the lines is connected to the VDD supply via a common pull-up resistor of 1.0Kohm in value. The I2CEN line can be configured as a push-pull output. The connection among master and many slave's devices is shown in figure #4. The number of devices can be connected to the M3L-bus is limited only by the maximum bus load capacitance of 200pF.
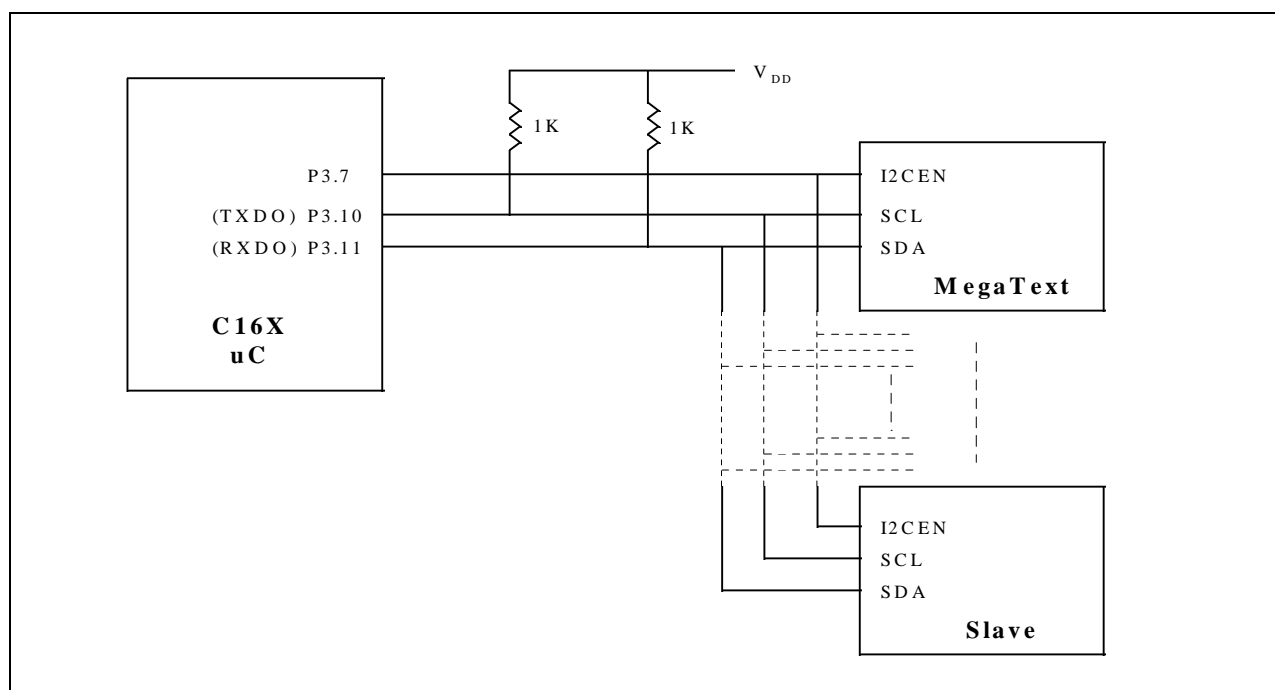

Figure 4:

Hardware connection among master and slave devices

3 Description of Software

3.1 Software Concept

The M3L-bus is generated by the Asynchronous/Synchronous Serial Channel(USART) of the C16X microcontroller. The clock frequency of the M3L-bus is 833 Khz with 20 Mhz CPU of the microcontroller. The baud rate for synchronous operation of the serial channel can be determined by the following formula:

$$B_{Sync} = \frac{f_{CPU}}{4 * (2+<SOBRS>)*(<SOBG>+1)}$$

$$SOBG = \left(\frac{f_{CPU}}{4 * (2+<SOBRS>)* B_{Sync}} \right) - 1$$

<SOBG> -- represents the content of the reload register.

<SOBRS> -- represents the value of bit SOBRS, in this case SOBRS =0.

The M3L_ASC.C software module is divided into 5 software subroutines which can be accessed by the main or external program. Those 5 software subroutines are used to construct the data transfer format of the M3L-bus which consists of two types; writing data to slave, and reading data from slave. Those 5 software subroutines are M3lInit, M3lStart, M3lMasterWrite, M3lMasterRead, and M3lStop.

The two types of data transfer format are written in the M3L_TEST.C. The M3L_TEST.C is a simple test program which just to verify the M3L_ASC.C software module. This test program is to transmit 6 bytes of data to MegaText IC from the array location of the microcontroller. The 6 bytes of data will be stored in the R8 to R13 of the MegaText IC. Next, the microcontroller will read back the contents of the 6 bytes from R8 to R13 of the MegaText IC and then store it into another array location of the microcontroller.

3.2   Description of Module Subroutines

M3L-BUS Software Module

Description

This module is a standard M3L-bus single master protocol by using
Asynchronous/Synchronous serial interface.

Module Subroutines

1. void CheckClockLine();
2. void M3lInit();
3. unsigned char M3lStart();
4. void M3lMasterWrite(unsigned char input_byte);
5. unsigned char M3lMasterRead();
6. void M3lStop();

void CheckClockLine()

Read the SCL line for wait condition. It will wait until the line has been released from
slave device.

Parameter	Description
None	

void M3lInit()

Initialize the port and serial communication. P3.7 - I2CEN, P3.10 - SCLK, P3.11 - SDA.
Configure the asynchronous/synchronous channel to synchronous mode and set the baud
rate to 833 KBaud. Pull-up resistors are required for SCL and SDA lines.

Parameter	Description
None	

unsigned char M3lStart()

Generate a start condition on M3L bus. Before that check the clock and data line for any bus faulty like no pull-up resistor on SDA/SCL or pull-down to low by the slave device. The time-out for bus faulty is approximately 10 ms before the start of M3L protocol.

Parameter	Description
None	

Return
The return value is "0" if the clock and data lines have no problem. Otherwise, the return value will be "1".

`void M3lMasterWrite(unsigned char input_byte)`

Output one byte of data to the slave device.

Parameter	Description
unsigned char input_byte	one byte of data to be sent to slave.

`unsigned char M3lMasterRead()`

Read one byte of data from the slave device.

Parameter	Description
None	

Return
Master device will receive one byte of data from the slave device.

`void M3lStop()`

Check for any WAIT condition before generating a STOP condition on the M3L-bus.

Parameter	Description
None	

M3L-BUS Application Software

Source file: M3L_TEST.C
Header file: M3L.H

SIEMENS

Description

This main program is to transmit 6 bytes of data to MegaText (SDA 5273) from the "raw_data" array. The 6 bytes of data will be stored in the R8 to R13 of MegaText IC. The device address of SDA 5273 is MSB (R/W)1000100 LSB. In M3L bus mode, the LSB of a byte is transmitted first. The sub-address of R8 is 00001000B. Next, the microcontroller will read back the contents of the 6 bytes from R8 to R13 and store it in the "stored_data" array.

Software subroutines

1. unsigned char WriteMegatext(unsigned char sub_addr,unsigned char *buffer,unsigned char num_byte)
2. unsigned char ReadMegatext(unsigned char sub_addr,unsigned char *buffer,unsigned char num_byte)

```
unsigned char WriteMegatext(unsigned char sub_addr,unsigned char
*buffer,unsigned char num_byte)
```

Write number of data bytes to MegaText. The flow of this subroutine is derived from the data format of writing to the MegaText as in the figure #1. If there is a bus faulty, the data will not be written.

Parameter	Description
unsigned char sub_addr	specifies the sub-address
unsigned char *buffer	point to the addressed location
unsigned char num_byte	number of bytes to be written

Return
The return value is "0" if the clock and data lines have no problem. Otherwise, the return value will be "1".

```
unsigned char ReadMegatext(unsigned char sub_addr,unsigned char
*buffer,unsigned char num_byte)
```

Read number of bytes from MegaText. The flow of this subroutine is derived from the data format of reading from the MegaText as in the figure #2. If there is a bus faulty, the data will not be read.

Parameter	Description
unsigned char sub_addr	specifies the sub-address
unsigned char *buffer	point to the addressed location
unsigned char num_byte	number of bytes to be read

Return

The return value is "0" if the clock and data lines have no problem. Otherwise, the return value will be "1".

3.3   Compilation of Software

The compilation of this software is using the KEIL C166 compiler. First of all, under the PROJECT, clicks on the "New Project", then key in the name of this project and add files to the project which are the M3L_ASC.C and M3L_TEST.C. Then, save the project. After-that, go to the OPTIONS and click on the "C166 Compiler..." . Lastly, select the option under OBJECT and cross the box under "Enable 80C167 instructions". This option will allow you to use the C16X derivatives. Now the project is really to compile and link all the object files. The compiling and linking of the project can be done by clicking the icon "BUILD ALL".

The AP162301.EXE is a compressed file contains M3L.H, M3L_ASC.C, and M3L_TEST.C. All these files are necessary to complete the compilation of the software program.