NB3H5150 I2C Programming Guide

I2C/SMBus Custom Configuration Application Note

3/14/2016

Table of Contents

Introduction

This application note provides details on configuring the NB3H5150 by writing to registers in the NB3H5150 through the I2C/SMBus interface.

Overview Process of Configuring a GUI style register file for the NB3H5150 and programming this file into the device registers via the I2C/SMBus.

The user will select the following:

Desired Register file from ON Semiconductor Website or from the NB3H5150 GUI register file folder.

Standard I2C Communication Protocol

Byte Read/Write

Reading or writing a register in a SMBus slave device in byte mode always involves specifying the register number.

Read. The standard byte read is as shown in the following figure. It is an extension of the byte write. The write start condition is repeated then the slave device starts sending data and the master acknowledges it until the last byte is sent. The master terminates the transfer with a NAK, then a stop condition. For byte operation, the 2*7th bit of the command byte must be set. For block operations, the 2*7th bit must be reset. If the bit is not set, the next byte must be the byte transfer count.

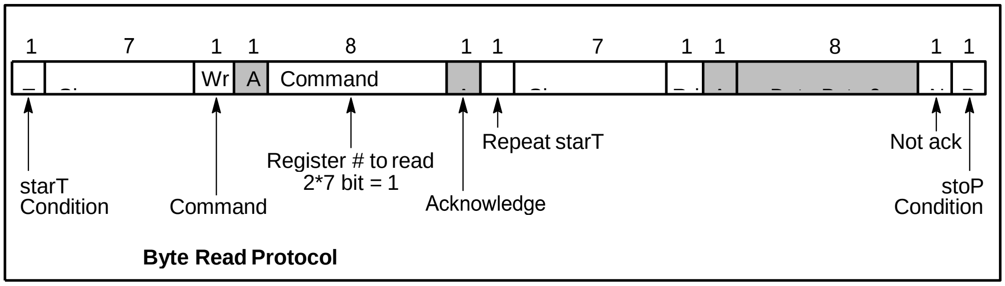


Figure 1. Byte Read Protocol

Write. The following figure illustrates a simple typical byte write. For byte operation the 2*7th bit of the command byte must be set. For block operations, the 2*7th bit must be reset. If the bit is not set, the next byte must be the byte transfer count. The count can be between 1 and 32. It is not allowed to be zero or exceed 32.

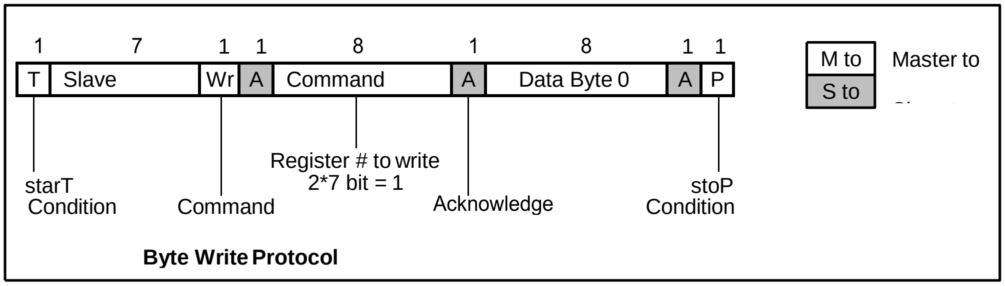


Figure 2. Byte Write Protocol

Block Read/Write

Read. After the slave address is sent with the r/w condition bit *set*, the command byte is sent with the MSB = 0. The slave Ack's the register index in the command byte. The master sends a repeat start function. After the slave Ack's this, the slave sends the number of bytes it wants to transfer (>0 and <33). The master Ack's each byte except the last and sends a stop function.

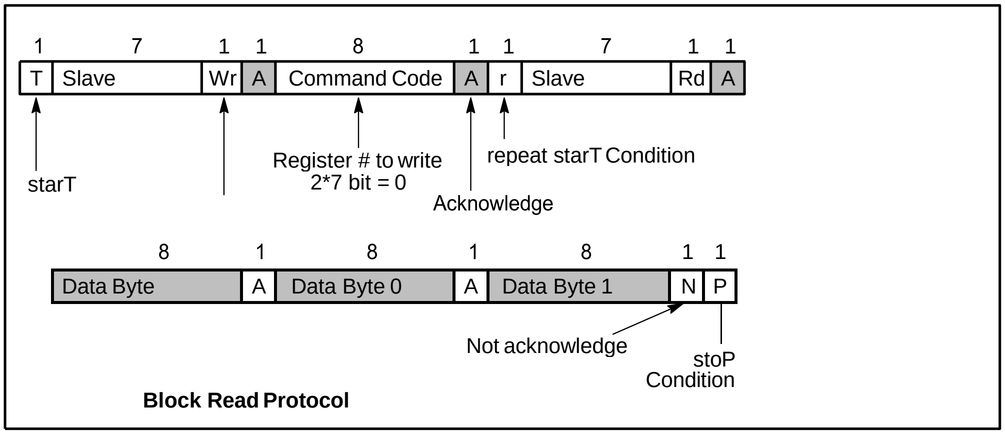


Figure 3. Block Read Protocol

Write. After the slave address is sent with the r/w condition bit not set, the command byte is sent with the MSB = 0. The lower seven bits indicate what register to start the transfer at. If the command byte is 00h, the slave device will be compatible with existing block mode slave devices. The next byte of a write must be the count of bytes that the master will transfer to the slave device. The byte count must be greater than zero and less than 33. Following this byte are the data bytes to be transferred to the slave device. The slave device always acknowledges each byte received. The transfer is terminated after the slave sends the Ack and the master sends a stop function.

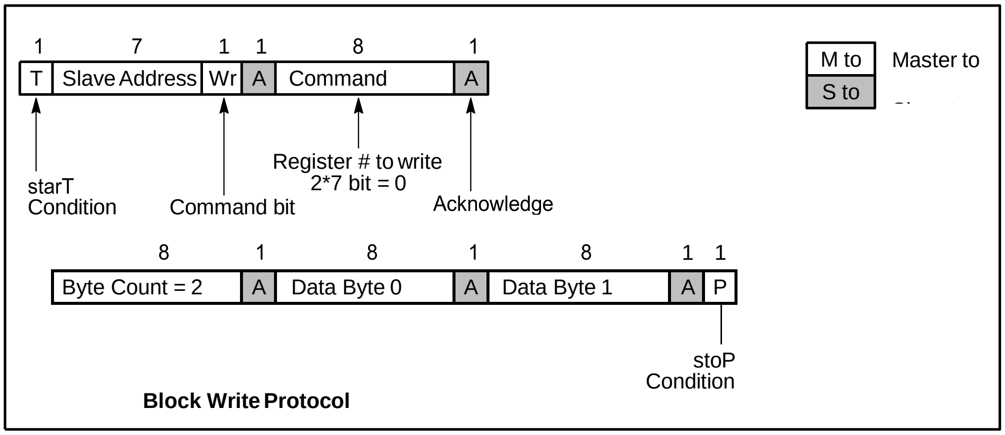


Figure 4. Block Write Protocol

I2C/SMBus Programming & Operation with NB3H5150

SDA/SCL Serial Hardware Interface with NB3H5150

The NB3H5150 provides an I2C / SMBus digital interface that can be used with any standard 2-wire I2C host device. Configuration, control and operation of the NB3H5150 are handled by reading and writing to the RAM space using the I2C/SMBus interface.

The I2C bus consists of a bidirectional serial data line (SDA) and a serial clock input (SCL) as shown in Figure 9 . In addition, the NB3H5150 operates as a slave device on the 2-wire SDA/SCL serial bus, compatible with the popular SMBus version 2.0 or I2C specification.

To enable the NB3H5150 to be in I2C/SMBus mode, both the SDA and SCL/PD pins must be connected to the VDD supply via external pull-up resistors on the I2C/SMBus. Recommend 10k-Ω.

The NB3H5150 operates in Standard-Mode transfer up to 100 kbps and supports 7-bit addressing.

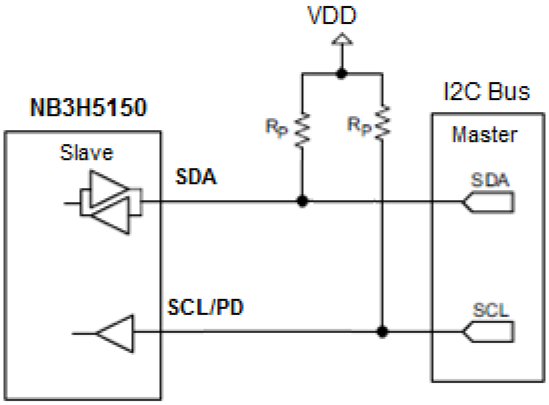


Figure 9. SDA / SCL Hardware Inteface

Device Slave Address Selection for NB3H5150

When communicating with multiple I2C/SMBus devices using the I2C/SMBus interface, each device must have its own unique address so the host can distinguish between the devices. Only the device with a matching slave address responds to subsequent I2C commands.

The device whose address corresponds to the transmitted address will respond by sending an acknowledge bit. All other devices on the bus remain idle while the selected device waits for data transfer with the master. These slave devices are accessed via a 7-bit slave address transmitted as part of an I2C packet.

The NB3H5150 allows up to **nine** unique NB3H5150 slave devices to be used in the system in addition to any other I2C slave device with a different I2C address. Slave addresses for the NB3H5150 are shown in Table 2. The NB3H5150 slave address is defined and can be set according to the pin-strap options listed in Table 8 of the datasheet and is set when the MMC pin is Low.

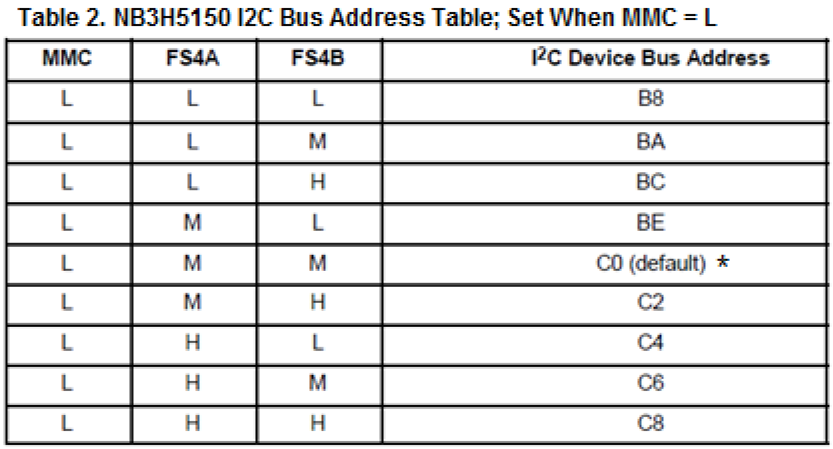


Table 2. NB3H5150 I2C Bus Address Table; Set When MMC = L

MMC	FS4A	FS4B	I^2C Device Bus Address
L	L	L	B8
L	L	M	BA
L	L	H	BC
L	M	L	BE
L	M	M	C0 (default) *
L	M	H	C2
L	H	L	C4
L	H	M	C6
L	H	H	C8

* FS4A and FS4B float to Mid-level when open.

User Programming Procedure

SMBus / I2C selection

The NB3H5150 (by default on power-up) is in SMBus mode.

The bit to change between SMBus and I2C modes is located in Register 0x08 bit 5.

Do not change any other bits in byte address 0x08 other than Bit 5. The other bits are reserved and must never be set to logic 1.

1

Byte	0x08

Bit	Description	If Bit = 0	If Bit =1	Type	Default
0	Reserved				0
1	Reserved				0
2	Reserved				0
3	Reserved				0
4	Reserved				0
5		SMBus mode	I2C mode		0,SMBus
6	Reserved				0
7	Reserved				0

Bank Switching Operation

I2C can only access 256 addresses directly (i.e. 0x00 through 0xFF).
SMBus can only access 128 addresses directly (i.e. 0x00 through 0x7F).
NB3H5150 has an address space which goes up to 0x14F. Therefore, a **“bank switching”** mechanism is included to allow access to all addresses no matter which bus protocol is used. Effectively, the two bank select bits are appended to the most-significant end of the seven-bit byte offset, creating a nine-bit address.
The bank switch control bits(6:7) are located in Register 0x21.
The most significant two bits of that register are used for bank selection; the least significant six bits are used for the READBYTECOUNT Function.

A provision has been made to always be able to access the register which controls the bank selection, no matter which bank is currently selected. An SMBus byte mode operation on the register (SMBUSCTL), or an I2C operation on the 0x21 register, where that register is the first (or only) byte of the operation will unconditionally be routed to the 0x21 register, no matter what the bank select bits are.
Conversely, this means that Read/Write performed with SMBus block-mode operations, or Read/Writes on all but the first byte of a multi-byte I2C operation will use the bank select bits in forming the effect address.

NOTE: It is not possible to cross from one bank to another in a block mode operation.

SMBus Mode Address Bank Selection:

Byte 0x21, Bits 7:6	SMBus Mode Bank	SMBus Byte Offset	NB3H5150 Registers Accessed
00	1	0x00-0x7F	0x00-0x7F
01	2	0x00-0x7F	0x80-0xFF
10	3	0x00-0x7F	0x100-0x17F
11	4	0x00-0x7F	0x180-0x1FF

I2C Mode Address Bank Selection:

Register 0x21, Bits 7:6	I2C Mode Bank	I2C Byte Offset	NB3H5150 Registers Accessed
0x	1	0x00-0xFF	0x00-0xFF
1x	2	0x00-0xFF	0x100-0x1FF

READBYTECOUNT bits

The least-significant bits in the SMBUSCTL register are for the READ byte count. This controls the number of bytes of data that the device will return on an I2C/SMBus Block Read operation. Typically, these six bits would be set to 0x20 (32 decimal) which is the maximum allowed by the SMBus protocol. The SMBus master can read fewer than this number in a block read without any adverse consequences.

Electrical and Timing Specifications

AC and DC electrical specifications for the SCL and SDA pins are shown in Table 3.
The timing specifications and timing diagram for the I2C bus are compatible with the I2C-Bus Standard.

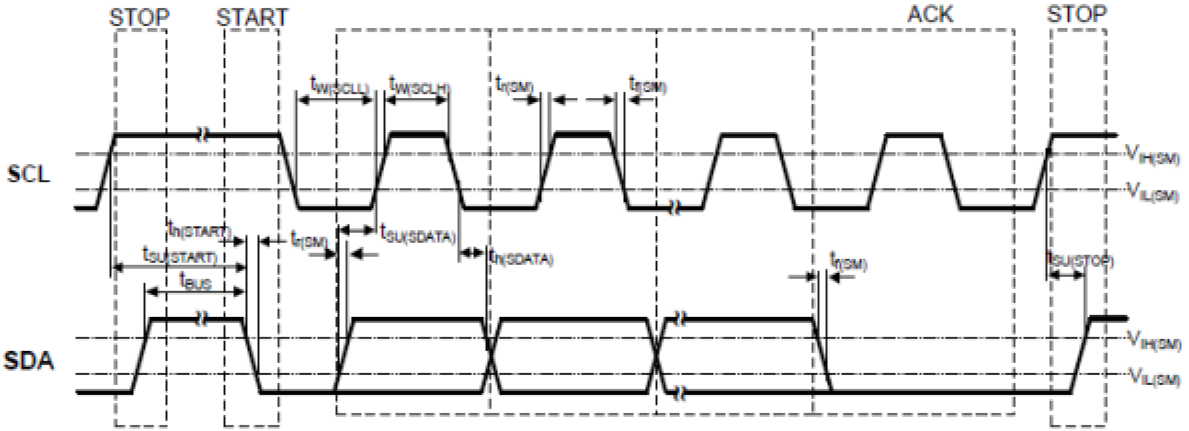
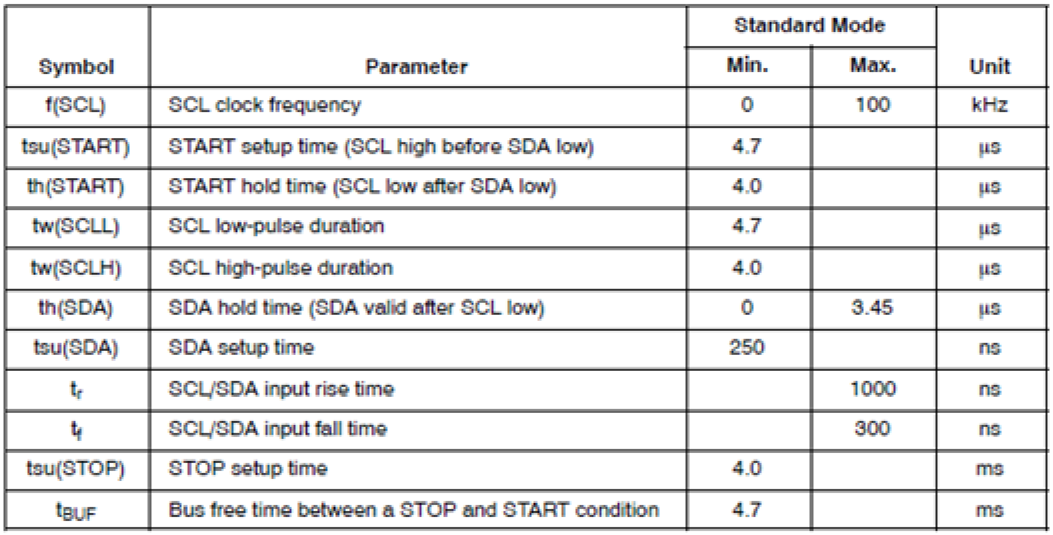


Figure 11. I²C Timing Diagram

Table 3. SDA/SCL TIMING REQUIREMENTS



Symbol	Parameter	Standard Mode		Unit
		Min.	Max.	
f(SCL)	SCL clock frequency	0	100	kHz
tsu(START)	START setup time (SCL high before SDA low)	4.7		µs
th(START)	START hold time (SCL low after SDA low)	4.0		µs
tw(SCLL)	SCL low-pulse duration	4.7		µs
tw(SCLH)	SCL high-pulse duration	4.0		µs
th(SDA)	SDA hold time (SDA valid after SCL low)	0	3.45	µs
tsu(SDA)	SDA setup time	250		ns
t_r	SCL/SDA input rise time		1000	ns
t_f	SCL/SDA input fall time		300	ns
tsu(STOP)	STOP setup time	4.0		ms
t_{BUF}	Bus free time between a STOP and START condition	4.7		ms

1