Application Note AN0031

I2C NMEA

For

Venus 8 GNSS Receiver

Ver 3

May 22, 2014

Introduction

Skytraq I2C slave interface provides two communication channels between I2C master and slave, the “TX” channel is used to convey NMEA message from i2C slave (Skytraq GNSS receiver) to master. The “RX” channel is used to convey binary message or command from I2C master to slave.

SkyTraq I2C Slave Registers

I2C SLAVE registers

Address	Name	Type	Function
0x0	GPREG0	R/W	General purpose register 0 , 8 bits
0x1	GPREG1	R/W	General purpose register 1 , 8 bits
0x2	GPREG2	R/W	General purpose register 2 , 8 bits
0x3	GPREG3	R/W	General purpose register 3 , 8 bits
0x4	GPREG4	R/W	General purpose register 4 , 8 bits
0x5	GPREG5	R/W	General purpose register 5 , 8 bits
0x6	GPREG6	R/W	General purpose register 6 , 8 bits
0x7	GPREG7	R/W	General purpose register 7 , 8 bits
0x8	CTRL	R/W	Control register

GPREG 0x0 ~ 0x7

Field	Bits	Reset	Description
DATA	7:0	UNDEF	General purpose data registers, GPREG0x0 ~ 0x4 are used as TX buffer, GPREG0x5~0x7 are used as RX buffer.

CONTROL 0x8

Field	Bits	Reset	Description
TX_DATA_SZ	7:5	UNDEF	Indicates to I2C master the valid data bytes in TX buffer (for master to read). GPREG0 ~4 are used as TX buffer, the valid data bytes thus range between 0~5.
RX_DATA_SZ	4:2	UNDEF	Written by I2C master to indicate the valid data bytes in RX buffer (for slave to read). GPREG5 ~7 are used as RX buffer, the valid data bytes thus range between 0~3.
TX_BUF_RDY	1	0	TX_BUF_RDY is a handshake bit used for TX flow control. This bit is set to 1 by I2C slave when the TX buffer is ready for I2C master to read. I2C master should clear this bit to 0 after reading the data in TX buffer. The valid data bytes in TX buffer is indicated in TX_DATA_SZ.
RX_BUF_RDY	0	0	RX_BUF_RDY is a handshake bit used for RX flow control. This bit is set to 1 by I2C master when the RX buffer is ready for I2C slave to read. I2C slave should clear this bit to 0 after reading the data in RX buffer. The valid data bytes in RX buffer is indicated in RX_DATA_SZ.

SkyTraq I2C Slave Implementation

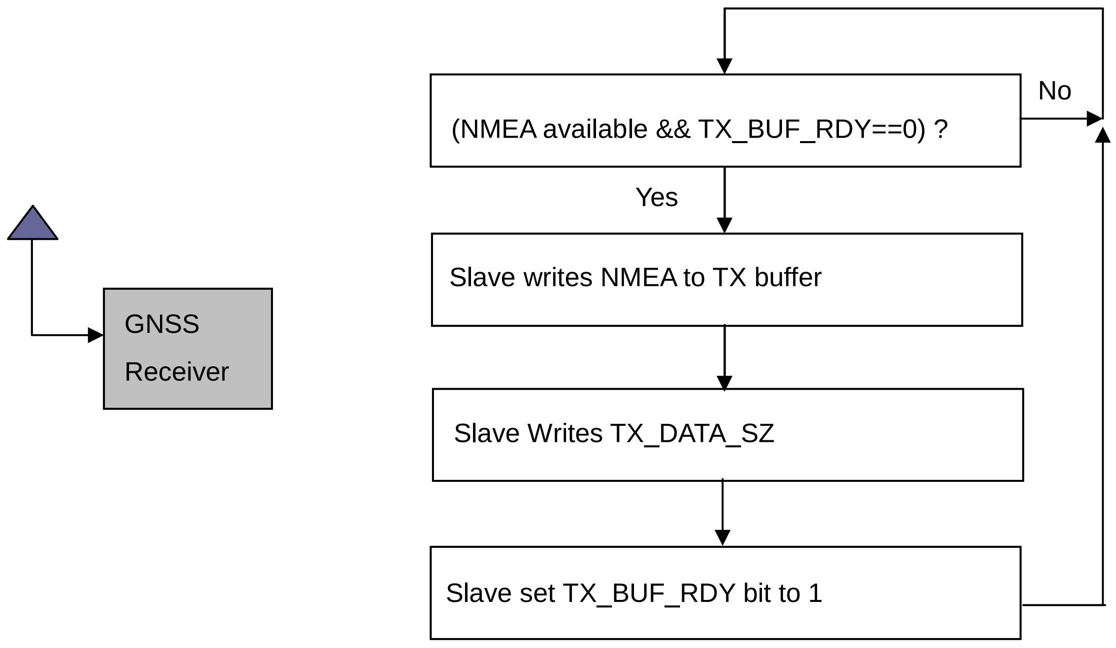
Data communication between I2C Master and Slave can be categorized in two ways – TX and RX. In Skytraq's I2C Slave implementation, we define TX as "Master read, Slave write" direction, and RX as "Master write, Slave read" direction.

In the TX direction, Master polls TX_BUF_RDY to determine if there are NMEA messages available, if TX_BUF_RDY is 1, Master will then read TX_DATA_SZ to determine the bytes counts. NMEA data are stored in GPREG0 to GPREG4. After Master reading the NMEA message, TX_BUF_RDY should be cleared to 0 by Master to inform Slave of the TX buffer being read.
Similarly, in the Slave side, Slave internally polls the status of TX_BUF_RDY, if TX_BUF_RDY is 0 and there are NMEA to be sent to Master, Slave will write NMEA data to GPREG0~4 and set TX_DATA_SZ, and then set TX_BUF_RDY to 1.
Eg, when Master polls TX_BUF_RDY=1 and TX_DATA_SZ=3, then Master can read 3 bytes of NMEA messages from GPREG0 to GPREG2. After completing reading, Master should then set TX_BUF_RDY to 0.

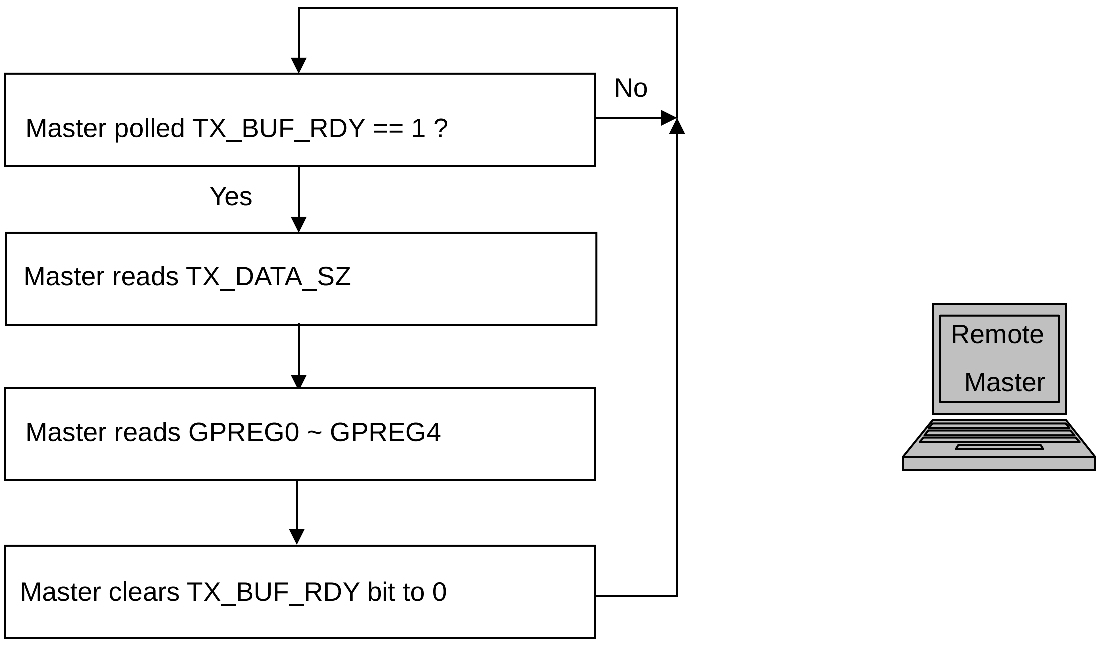
In the RX direction, If master has binary message/command for Slave and RX_BUF_RDY is 0, then Master may write binary message/command to GPREG5~7 and set RX_DATA_SZ, then set RX_BUF_RDY to 1 to inform Slave of the incoming RX data.
The transition of RX_BUF_RDY from 0 to 1 will trigger an interrupt in the Slave side. On the interrupt, Slave read RX_DATA_SZ and RX data (GPREG5~7), after reading, Slave clears RX_BUF_RDY to 0.
Eg. When master has 3 bytes of binary command for Slave, and Master polled RX_BUF_RDY to be 0, then Master may write RX_DATA_SZ=3 and 3 bytes to GPREG5~7. After then, Master set RX_BUF_RDY to 1 to inform Slave of the incoming RX data.

I2C NMEA Master and Slave Flow

Case I: I2C Master reads NMEA from I2C Slave

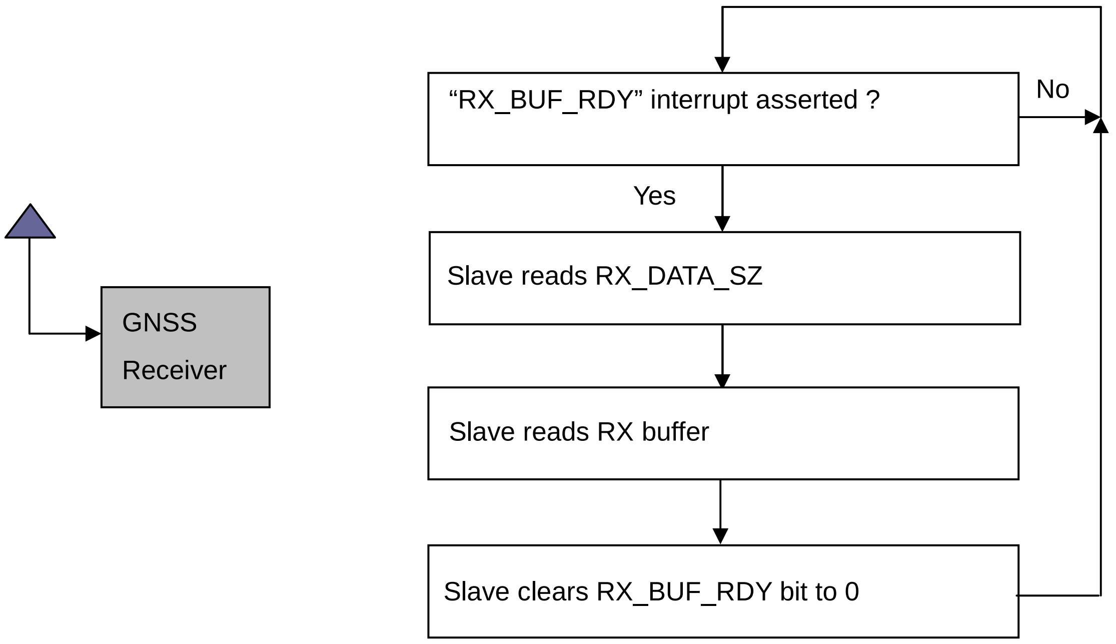


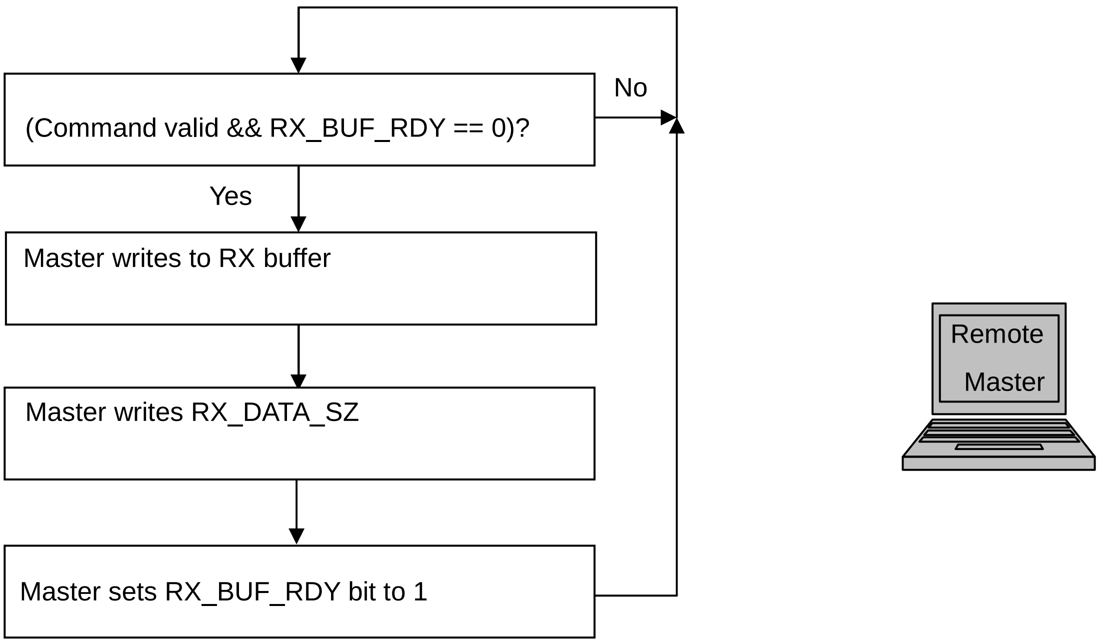
(NMEA available && TX_BUF_RDY==0) ?
No
Yes
Slave writes NMEA to TX buffer
Slave Writes TX_DATA_SZ
Slave set TX_BUF_RDY bit to 1
GNSS
Receiver



Master polled TX_BUF_RDY == 1 ?
No
Yes
Master reads TX_DATA_SZ
Master reads GPREG0 ~ GPREG4
Master clears TX_BUF_RDY bit to 0
Remote
Master

Case II: I2C Master write binary message/command to I2C Slave


GNSS
Receiver
"RX_BUF_RDY" interrupt asserted ?
No
Yes
Slave reads RX_DATA_SZ
Slave reads RX buffer
Slave clears RX_BUF_RDY bit to 0


(Command valid && RX_BUF_RDY == 0)?
No
Yes
Master writes to RX buffer
Master writes RX_DATA_SZ
Master sets RX_BUF_RDY bit to 1
Remote
Master

I2C Example

Read data from Control Register (refer to Sample Codes and I2C-Master Core Specification)

I2C Slave Read sequence (generated from Master)

1. Generate a start (STA) + slave address + write bit (WR) signal
   - Wait for RxACK and TIP in Status Register
2. Generate a write slave register address signal
   - Wait for RxACK and TIP in Status Register
3. Generate a start (STA) + slave address + read bit (RD) signal
   - Wait for RxACK and TIP in Status Register
4. Generate a stop (STO) + Read (RD) + no ack (NACK) signal
   - Wait for TIP in Status Register
5. Read data from receive register

So, the pseudo-code will be as follows,

```
#DEFINE VENUS_I2C_WR_ADDR       0x78
#DEFINE VENUS_I2C_RD_ADDR      (VENUS_I2C_WR_ADDR+1)
#DEFINE I2C_SLAVE_CONTROL_REG    8         //CONTROL REGISTER

//polling slave control register status
if(i2c_write_start(VENUS_I2C_WR_ADDR)==TRUE)
{
   if(i2c_write(I2C_SLAVE_CONTROL_REG)==TRUE)//set control register address
   {
      if(i2c_write_start(VENUS_I2C_RD_ADDR)==TRUE)
      {
          rdy=i2c_read(1);
      }
      else
      return;//error
   }
   else
   return;
}
else
return;
```

Change Log

Ver 3, May 22, 2014

1. Add **“I2C Example”** section.

Ver 2, March 6, 2014
1. Added timing figure

Ver 1 Dec. 5, 2013

1. Initial release

SkyTraq Technology, Inc.
4F, No.26, Minsiang Street, Hsinchu, Taiwan, 300

Phone:	+886 3 5678650
Fax:	+886 3 5678680
Email:	info@skytraq.com.tw