Keywords: high speed interconnect, SerDes, serializer, deserializer, I²C, GMSL, automotive video, gigabit multimedia serial link

APPLICATION NOTE 5064

Access 16-Bit I²C Peripheral Register Addresses Through the GMSL SerDes Remote-Side I²C Interface

By: Alex Shih

Jun 20, 2011

Abstract: This application note describes how to access 16-bit register addresses through the gigabit multimedia serial link (GMSL) SerDes' remote-side I²C interface.

Introduction

The Maxim gigabit multimedia serial link (GMSL) serializer/deserializer (SerDes) family, which includes the MAX9249, MAX9259, MAX9260, MAX9263, and MAX9264, has a UART/I²C converter available at the remote side to provide support for I²C interfaced peripherals. These peripherals include camera sensors, I/O port expanders, audio analog-to-digital converters (ADCs) and digital-to-analog converters (DACs), and LED lighting controls.

In the GMSL data sheets and related technical documents, the side that directly connects to the electronic control unit (ECU) through the UART port is referred to as the "local side" and the other side is referred to as the "remote side."

The I²C peripheral interface is the default setting in the control-channel's base mode. The GMSL family SerDes only converts UART packets that have a device address different from those of the GMSL serializer or deserializer. The UART/I²C converter accepts UART read or write packets issued, for example, by an automotive ECU and converts them to an I²C master protocol.

The I²C interface supports various formats that the peripheral may use, with or without index registers and with 8-bit or 16-bit index register's address, accomplished with the remote-side register bit I2CMETHOD (register 0x05 bit 7) settings.

1. For a peripheral without index registers:

   Set I2CMETHOD = 1, following the standard GMSL UART protocol to send the read/write packet. The UART/I²C converter will remove the 8-bit Register Address in the read/write packet as illustrated in **Figure 1**.

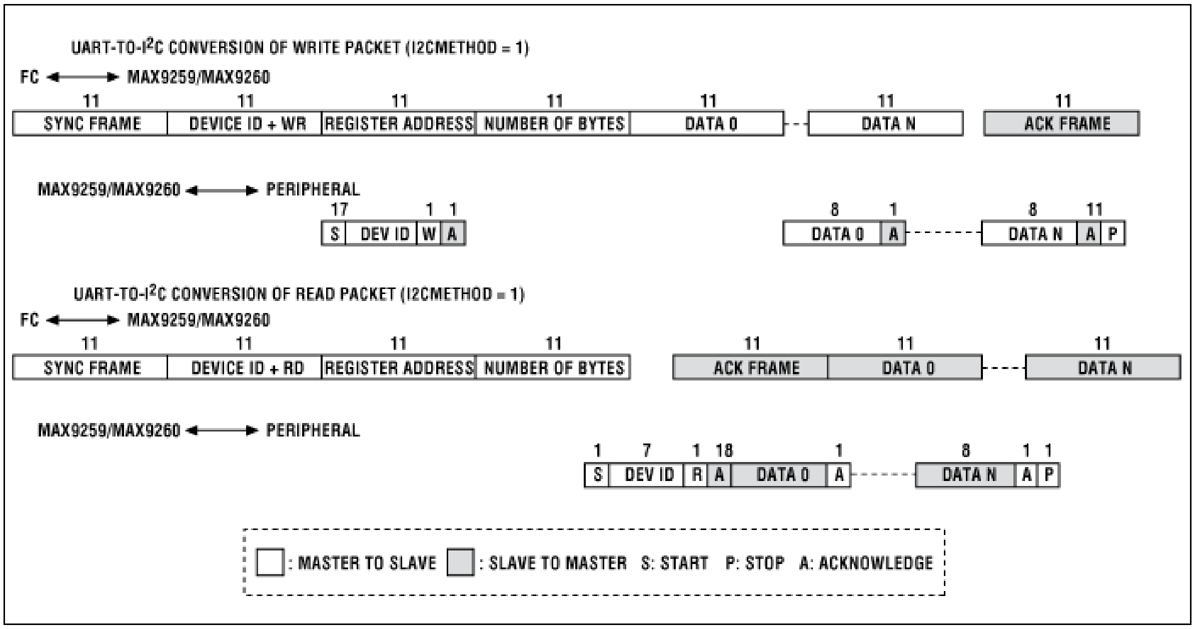


Figure 1. Format conversion between UART and I²C in Command-Byte-Only Mode (I2CMETHOD = 1).

2. For a peripheral with 8-bit address index registers:

   Set I2CMETHOD = 0, following the standard GMSL UART protocol to send the read/write packet, as shown in **Figure 2**.

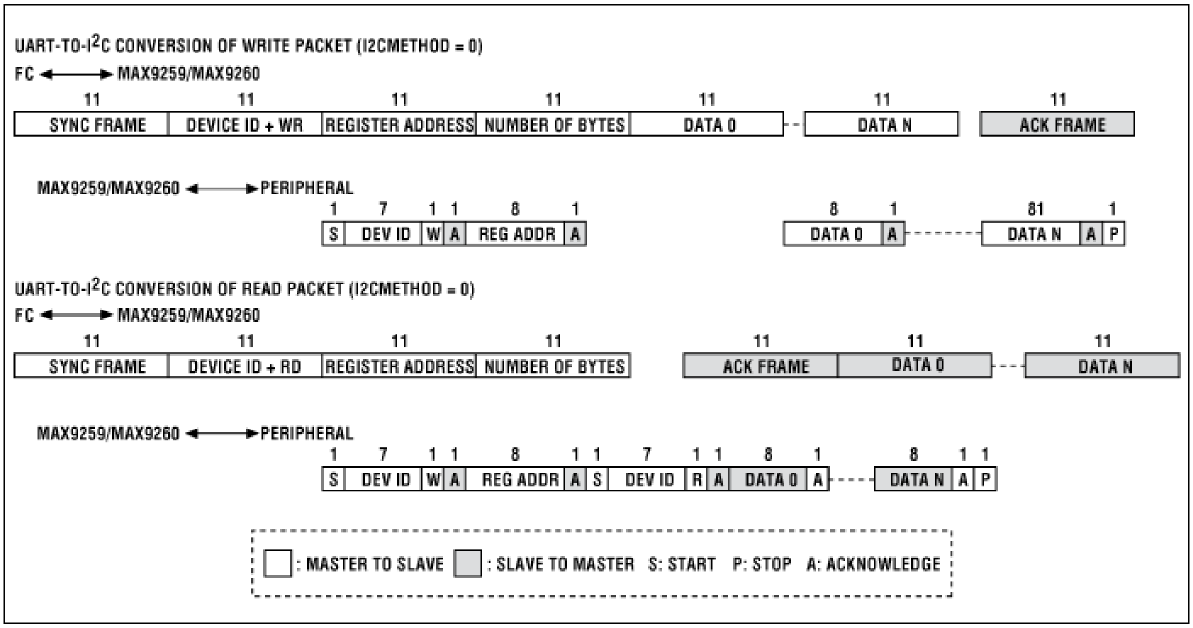
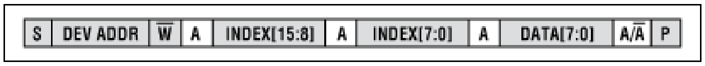


Figure 2. Format conversion between UART and I²C with Register Address (I2CMETHOD = 0).

3. For a peripheral with 16-bit address index registers:

   3.1. **Write Operation**. Set I2CMETHOD = 1 and follow the standard GMSL UART protocol to send the write packet. Put the 16-bit index register address in the first two bytes of data and make the Number of Bytes equal to the original number of bytes needing to be sent plus two. The UART/I²C converter will remove the 8-bit Register Address in the standard write packet as illustrated in the Figure 1.

For example, if the peripheral requires a 16-bit index register address and 8-bit data like this:

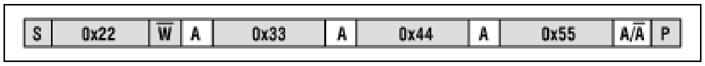


S	DEV ADDR	$\overline{W}$	A	INDEX[15:8]	A	INDEX[7:0]	A	DATA[7:0]	A/$\overline{A}$	P

and device_address = 0x22, index_register_address = 0x3344, and data = 0x55. The host ECU needs to write the packet 0x79 + 0x22 + 0x00 + 0x03 + 0x33 + 0x44 + 0x55 to the UART port, and then read the ACK byte from the UART port to check if the write operation successful.

0x79 is the sync byte,
0x22 is the device address with write operation indication,
0x00 is a dummy byte and could be any value; the UART/I²C converter will remove it during the conversion.
0x03 is the number of bytes,
0x33 is the high byte of 16-bit index register address,
0x44 is the low byte of 16-bit index register address,
0x55 is the 8-bit data.

After the UART/I²C converter, the packet will become:



S	0x22	$\overline{W}$	A	0x33	A	0x44	A	0x55	A/$\overline{A}$	P

3.2. **Read Operation**. Two UART packets, a write packet followed by a read packet, are needed to accomplish the read operation. The write packet sets the index register address and the read packet reads the data.

Assuming the peripheral has the same device address and index register address, and one data byte needs to be read out, the following steps are needed:

Set I2CMETHOD = 1,
ECU sends the write packet 0x79 + 0x22 + 0x00 + 0x02 + 0x33 + 0x44 to the UART port,
ECU reads the ACK byte from UART port,
ECU sends the read packet 0x79 + 0x23 + 0x00 + 0x01 to the UART port,
ECU read the ACK byte from UART port,
and then ECU read the data from the UART port.

For the write packet:
0x79 is the sync byte,
0x22 is the device address with write operation indication,
0x00 is a dummy byte and could be any value; the UART/I²C converter will remove it during the conversion.
0x02 is the number of bytes,
0x33 is the high byte of 16-bit index register address,
0x44 is the low byte of 16-bit index register address.

For the read packet:
0x79 is the sync byte,
0x23 is the device address with read operation indication,
0x00 is a dummy byte and could be any value; the UART/I²C converter will remove it during the conversion,
0x01 is the number of bytes.

Please refer to the data sheet of Maxim's GMSL SerDes for detailed information on the UART protocol for base mode, sync byte, ACK byte, and the UART/I²C converter.

Related Parts		
MAX9249	Gigabit Multimedia Serial Link Serializer with LVDS System Interface	Free Samples
MAX9259	Gigabit Multimedia Serial Link with Spread Spectrum and Full-Duplex Control Channel	Free Samples
MAX9260	Gigabit Multimedia Serial Link with Spread Spectrum and Full-Duplex Control Channel	Free Samples

MAX9263	HDCP Gigabit Multimedia Serial Link Serializer/Deserializer	Free Samples
MAX9264	HDCP Gigabit Multimedia Serial Link Serializer/Deserializer	Free Samples
MAX9265	HDCP Gigabit Multimedia Serial Link Serializer with LVDS System Interface	Free Samples
MAX9266	HDCP Gigabit Multimedia Serial Link Deserializer with LVDS System Interface	Free Samples
MAX9268	Gigabit Multimedia Serial Link Deserializer with LVDS System Interface	
MAX9270	Gigabit Multimedia Deserializer with Spread Spectrum and Full-Duplex Control Channel	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 5064: http://www.maximintegrated.com/an5064
APPLICATION NOTE 5064, AN5064, AN 5064, APP5064, Appnote5064, Appnote 5064