Keywords: video, audio, serializer, deserializer, SerDes, GMSL, HDCP, high speed interconnect, automotive, communication, UART, I2C, control channel, LCD display, backup camera

APPLICATION NOTE 5535

# FAQs about the Control Channel and Operation of Serializer/Deserializer Devices

By: Wai Phyo

Apr 22, 2013

*Abstract: This application note addresses frequently asked questions (FAQs) about the control channel and operation of gigabit multimedia serial link (GMSL) serializer/deserializer (SerDes) devices. These products transport video and control information over low-cost shielded twisted pair (STP) or coax cables in vehicles. These questions are related to the serial link on these parts and the way it handles video data and control information.*

## Table of Contents

## Operation

### Compact GMSL Devices

### GMSL Devices

- How many errors cause the PLL to unlock?
- Why do GMSL devices have registers for serializer and deserializer device addresses?
- Can I reconfigure the bit order (such as LSB to MSB vs. MSB to LSB) at the deserializer?
- When the GMSL device is in sleep mode, can the µC access the remote device registers?
- What does the wake-up signal look like? Does it contain a specific pattern?
- Is it possible to directly connect the MAX9259 with a YUV422 image format? Or do you need a YUV-RGB data converter to input to a Maxim serializer?
- In most GMSL devices, the serial link data format—parallel input, DIN27—is allocated for the reserved bit (RES). Is this mapping the same in the MAX9263?
- What is sleep mode?
- What is a word boundary?
- How are incorrect serial-word boundaries detected? If an incorrect word boundary is detected, is it displayed as a parity error, decoding error, or both?
- What triggers loss of lock? In other words, if a parity error or decoding error is detected, how many parity or decoding errors will trigger the loss of lock?
- When bus contention occurs, will the MAX9263 and the MAX9264 register values change?
- Was the 9.6kbps lower rate implemented to support µCs that have a slower bit rate? Does the selected UART rate have any effect on the interrupt signal?

## Non-GMSL Devices

- Can the MAX9257 generate a STOP bit between the ACK and START conditions? If the µC sends two UART STOP bits, which side will eliminate the extra one? Will the $I^2C$ translator on the MAX9257 (peripheral) generate an extra STOP bit?
- What accuracy of baud rate is required for UART input to the MAX9258A for a 100kbps baud rate?

## Control Channel

- When the remote-side GMSL receive the wrong data packet, does the remote-side GMSL terminate the communication or does it process the wrong data packet to its output (UART or $I^2C$)?
- What is the maximum delay between frames of a control packet in UART mode?
- What is the wait time after a communication packet has finished before the next packet can be sent?
- What is the maximum time until the ACK frame should be received?
- How long is the control channel not available when MS is toggling?
- How long is the control channel not available when INT is toggling?
- How long is forward CC unavailable when switching from CLINKEN to SEREN or vice versa?
- What is the availability of reverse control channel after power-up?
- What is the availability of CC in case of wake-up via link?
- What is the CC behavior in the case of changing CDS at runtime?

## Configuration

- Can I reconfigure the bit order (such as LSB to MSB vs. MSB to LSB) at the deserializer?
- Is it possible to directly connect the MAX9259 with a YUV422 image format? Or do you need a YUV-RGB data converter to input to a Maxim serializer?
- My MAX9259/MAX9260 chipset is set with active-low AUTOS = low (SEREN = high and CLINKEN = low) in a camera application. After power-up, the camera provides a pixel clock, and the forward channel between the MAX9259 and the MAX9260 is active. I tried to write register 4 (0x04) to change SEREN = low and CLINKEN = high. I can only read the values of register 4, but I cannot write to register 4. There is no problem reading and writing to all other registers. Should I change active-low AUTOS = high to enter Configuration Link Mode at startup? Why can I read register 4 but I cannot write to it?
- In my setup, the control-direction selection (CDS) pin on the serializer is set to low and on the deserializer set to high, which means that each GMSL chip can connect to its respective µC. If the deserializer's register is set to INTTYPE = 01, then both sides of the µC are accessible to the SerDes register by the UART communication. Under this condition, will I need to provide a solution of my own to handle the bus contention-handling scheme? Does a bus fight exist between the µCs accessing the SerDes register?
- Does MS = L, CSD = L on the serializer and CDS = H on the deserializer make INTTYPE on both the serializer and the deserializer have a "01" setting?
- CDS = L and MS = L enable base mode by a UART. Is the INTTYPE setting of the serializer automatically set to "01" by this pin setting?
- What happens if spread-spectrum clocking is turned on at both sides? Is it possible that the buffer will be either over- or underflow?

## Operation

### Compact GMSL Devices

**Q) When the MAX9272 double output feature is not supported in the serializer, the data defaults to single input. What does this mean?**

A) If you pair the MAX9259 serializer with the MAX9272 deserializer, assume that the MAX9259 is single input. For example, if the MAX9259 and the MAX9272 are paired, BWS must be set the same on both ends. The MAX9272 can be in either single output or double output mode. The parallel data of MAX9259 DIN to MAX9272 DOUT mapping are as follows:

(BWS = 0, 24-bits mode) 9259 [DIN(0:20)] to 9272 [DRS = 0, DBL = 0; DOUT(0:21)]. 9272's DOUT21 will be zero. 9259 [DIN(0:20)] to 9272 [DRS = 0, DBL = 1; DOUTA(0:10)&DOUTB(0:10)]. 9272's DOUTB10 will be zero.

(BWS = 1, 3bits mode) 9259 [DIN(0:28)] to 9272 [BWS = 1, DBL = 0; DOUT(0:27)]. 9259's DIN28 will not be mapped. 9259 [DIN(0:28)] to 9272 [DRS = 0, DBL = 1; DOUTA(0:14)&DOUTB(0-14)]. 9272's DOUTB14 will be zero.

You can pair the GMSL MAX9259 and the compact GMSL MAX9272 device. Keep in mind that the MAX9272 has a pin limitation and an allowable PCLK frequency range that needs to be compatible with the MAX9259. Also, one or more features/functions supported in the MAX9259/MAX9272 are not supported in MAX9272/MAX9259 devices (e.g., audio, hamming coding, and CRC and HV encoding).

**Q) When the MAX9272 receives data from the MAX9271, do BWS and DBL require the same settings on both devices? For example, is it possible that the MAX9271 inputs data in single mode and the MAX9272 outputs data in double mode?**

A) Yes, the BWS settings must be the same on both sides. It is possible that the MAX9271 is in single input mode and the MAX9272 is in double output mode, or double at input and single at output.

The MAX9271 single input data will map to the MAX9272 double output as follows:

A. When BWS = 0, DIN(0:15) will map to DOUTA(0:10)&DOUTB(0:4). DOUTB(5:10) will be zero.

B. When BWS = 1, DIN(0:15) will map to DOUTA(0:14)&DOUTB(0). DOUTB(1:14) will be zero.

The MAX9271 double input data will map to the MAX9272 single output as follows:

A. When BWS = 0, DINA(0:10) will map to DOUT(0:10) and DINB(0:10) will map to DOUT(11:21).

B. When BWS = 1, DINA(0:12) will map to DOUT(0:12) and DINB(0:12) will map to DOUT(15:27).

DINA(13:14) and DINB(13:14) are a bit complicated. Either DINA(13:14) or DINB(13:14) will come out of DOUT(13:14), but we do not know which. Every time the system powers up, it can change. Therefore, if you want to use DIN(13:14), make DINA(13:14) and DINB(13:14) the same (i.e., half rate compared to DIN[0:12]).

**Q) Why does the MAX9272 have two HV outputs: HS0/VS0 and HS1/VS1?**

A) When DBL = 1 and HVEN = 1 on both sides, the MAX9271's input HS/VS values at the first DBL cycle are HS0/VS0 and the MAX9271's input HS/VS values at the second DBL cycle are HS1/VS1.

**Q) When using the MAX9271 in double mode, do I apply twice the frequency of PCLK in single mode to PLCLKIN to keep same transfer rate within single mode?**

A) Yes.

**Q) Does the MAX9271/MAX9272's description "25MHz to 75MHz clock for 15-bit double input/output" mean that the data transfer rate in double mode is half of the clock, 12.5M to 37.5M word/sec?**

A) Yes.

**Q) Is there a way for the serializer to drive a GPIO in a deserializer? What is the H/V encoding feature and what are its benefits?**

A) The GPIOs are connected to the register bit and they are not aligned with data. You can use any of the Dxx pins (if there are enough pins to transmit data + HSYNC + VSYNC) to transmit HSYNC and VSYNC. Then, HSYNC and VSYNC will be synchronized with pixels. In this case, HSYNC and VSYNC use two dedicated Dxx pins and continuously take up bandwidth of the serial link. Also, the HSYNC and VSYNC frequency is very slow. Enabling the H/V encoding maximizes the serial link BW but only D0:13 (14 data bits) will be available. H/V will use dedicated D14/15 in single-input mode.

## GMSL Devices

**Q) What is the recommended number for the error threshold (ERRTHR) register setting in a GMSL**

**device?**

A) There is no recommended or critical value. If the register is set to 0, each detected error will be reported. In most applications, a single error or a few errors may not matter and the system µC may need to only interfere if something more catastrophic occurs. In that case, you can set ERRTHR to a higher value and generate an error pulse only when many errors are detected.

**Q) What does Auto Error Reset (AUTORST) do and when is it needed?**

A) The Auto Error Reset function works as follows:

1. Set AUTORST = 1 in the deserializer.
2. Set a value to ERRTHR (Error Threshold) or keep it 0 as default.
3. The deserializer counts the number of detected errors and updates the DECERR register.
4. Whenever DECERR > ERRTHR, the deserializer will generate an approximately 1µs pulse at the ERRORB pin and then the DECERR register is reset to 0, so that DECERR <= ERRTHR again.
5. If more errors are detected, DECERR will increment again and Step 4 will be repeated.
6. When ERRTHR = 0 (default), an error pulse is generated at the ERRORB pin with each detected error.
7. When the Auto Error Reset function is used, ECU does not need to read the DECERR register to reset it, since it is automatically reset.

**Q) How many errors cause the PLL to unlock?**

A) Loss of lock is triggered by the type of error counting and filtering approach. With each word, if it has an error, a counter is incremented by 1. If a word does not have an error, the same counter is decremented but it never goes below 0. If the counter reaches a certain amount, then lock is lost. Therefore, loss of lock depends on the density of errors, not on the absolute number of errors. Under default conditions, if, on average, one out of four words has an error, the lock will be lost sooner or later.

**Q) Why do GMSL devices have registers for serializer and deserializer device addresses?**

A) Serializer and deserializer device addresses are needed for UART-to-I$^2$C conversion. When the device address of a packet does not match both the serializer and deserializer addresses, that packet is converted to I$^2$C. So, the serializer needs to know the deserializer address and vice versa. If a UART-to-I$^2$C converter is not used, the chip only needs to know its own address.

**Q) Can I reconfigure the bit order (such as LSB to MSB vs. MSB to LSB) at the deserializer?**

A) The bit orders are not internally configurable, but you can send video MSB from the LSB slot or vice versa. They can also tie MSB out of the deserializer to LSB on the display side.

**Q) When the GMSL device is in sleep mode, can the µC access the remote device registers?**

A) The GMSL deserializer does not accept any command or request while in sleep mode. It only responds to the wake-up signal.

**Q) What does the wake-up signal look like? Does it contain a specific pattern?**

A) The wake-up signal is a low-frequency clock-like pattern, approximately 550kHz. During sleep mode, the receiver only detects the wake-up signal. Although the wake-up signal consists of 15 cycles of

500kHz, the deserializer wakes up after three or four cycles have been detected.

**Q) Is it possible to directly connect the MAX9259 with a YUV422 image format? Or do you need a YUV-RGB data converter to input to a Maxim serializer?**

A) The GMSL device accepts RGB or YUV formats as digital inputs and serializes them.

**Q) In most GMSL devices, the serial link data format—parallel input, DIN27—is allocated for the reserved bit (RES). Is this mapping the same in the MAX9263?**

A) Yes, the RES bit is always transmitted from the bit allocated for DIN27.

**Q) What is sleep mode?**

A) The serializer/deserializer chipset includes a low-power sleep mode to reduce power consumption of the device not attached to the µC (e.g., the MAX9260 in LCD applications and the MAX9259 in camera applications). Set the corresponding remote IC's SLEEP bit to 1 to initiate sleep mode. The MAX9259 sleeps immediately after setting SLEEP = 1. The MAX9260 sleeps after serial link inactivity or 8ms (whichever arrives first) after setting SLEEP = 1. See the Link Startup Procedure section in the MAX9259/MAX9260 data sheet for details on waking up the device for different µC and starting conditions. The µC-side device cannot enter into sleep mode, and its SLEEP bit remains at 0. Use the PWDN input pin to bring the µC-side device into a low-power state.

**Q) What is a word boundary?**

A) A word boundary is how we segment the bits on the serial link into words and process them.

**Q) How are incorrect serial-word boundaries detected? If an incorrect word boundary is detected, is it displayed as a parity error, decoding error, or both?**

A) If a word boundary is not detected, there will be decoding and parity errors in a short time and the deserializer will lose lock and then search for a new word boundary. If it cannot find any valid word boundary, it will reset the CDR and try relocking.

**Q) What triggers loss of lock? In other words, if a parity error or decoding error is detected, how many parity or decoding errors will trigger the loss of lock?**

A) The GMSL deserializers have a counter to detect loss of lock. When there is a decoding or parity error, the counter is incremented. When there is no error, it is decremented by 1. (It cannot go below 0.) If the counter reaches the maximum threshold, the deserializer declares loss of lock and the LOCK pin goes low.

**Q) When bus contention occurs, will the MAX9263 and the MAX9264 register values change?**

A) When bus contention occurs, the MAX9263 and the MAX9264 register values will remain the same. However, the UART communication will be lost or corrupted. However, it is very unlikely that bus contention would cause any change in register settings.

**Q) Was the 9.6kbps lower rate implemented to support µCs that have a slower bit rate? Does the selected UART rate have any effect on the interrupt signal?**

A) Yes, the MAX9259/MAX9260 chipset was originally designed for a minimum of 100kbps. However, some µCs needed a slower bit rate, so we changed the design to support rates down to 9.6kbps. When slow UART rates are used, they can be confused as an INT signal on the link, so we do not support interrupts when the UART rate is below 100kbps in MAX9259/MAX9260 devices.

## Non-GMSL Devices

**Q) Can the MAX9257 generate a STOP bit between the ACK and START conditions? If the µC sends two UART STOP bits, which side will eliminate the extra one? Will the I$^2$C translator on the MAX9257 (peripheral) generate an extra STOP bit?**

A) Unfortunately, the MAX9257 cannot generate a STOP bit at this desired position. According to the I$^2$C protocol, repeated START bits are allowed (e.g., if a STOP condition is not needed). Two or more UART STOP bits are considered as a single STOP bit by the internal circuit. The number of STOP bits is not counted, so the I$^2$C translator will always generate a single STOP condition on the I$^2$C bus for each UART packet, regardless of how many STOP bits are used after each UART frame. However, GMSL devices can generate a STOP at that position by using the I$^2$CMETHOD = 1 setting and two separate packets: one write and then one read.

**Q) What accuracy of baud rate is required for UART input to the MAX9258A for a 100kbps baud rate?**

A) The UART baud rate accuracy should be higher than 5%. For example, for 100kbps, we need 9.5µs < bit time < 10.5µs for each UART bit.

## Control Channel

**Q) When the remote-side GMSL receive the wrong data packet, does the remote-side GMSL terminate the communication or does it process the wrong data packet to its output (UART or I$^2$C)?**

A) When the remote side is I$^2$C, the remote-side GMSL device terminates the control channel packet when a parity error is detected. When the remote side is a UART, whatever comes in goes through to the peripherals. A remote-side slave checks for errors and terminates the packet.

**Q) What is the maximum delay between frames of a control packet in UART mode?**

A) The maximum delay between frames of a control packet should be 2 bit times.

**Q) What is the wait time after a communication packet has finished before the next packet can be sent?**

A) It is 16 bit times after the end of the packet before a new packet can be started.

**Q) What is the maximum time until the ACK frame should be received?**

A) The GMSL will send the ACK frame not more than 4 bit times after the STOP bit of the previous frame. One UART frame is 11 bits long and may experience some link delays. If the ACK frame is not received in 16 bit times after the end of (STOP bit of) the last frame, then assume that either it was not sent or there is a problem with the control channel link (e.g., CC is disabled).

**Q) How long is the control channel not available when MS is toggling?**

A) When MS of the deserializer is toggled, CC is not available for 1ms. This unavailability takes effect immediately. Note that toggling the serializer MS does not disable the control channel.

**Q) How long is the control channel not available when INT is toggling?**

A) When INT of the deserializer is toggled, CC is not available for 350µs.

**Q) How long is forward CC unavailable when switching from CLINKEN to SEREN or vice versa?**

A) Add the serializer's lock time and the deseralizer's link start up time.

**Q) What is the availability of reverse control channel after power-up?**

A) Reverse control channel is available at the serializer after the serializer has powered up. But when the serial link starts or stops, reverse control channel is temporarily disabled for 350µs.

**Q) What is the availability of CC in case of wake-up via link?**

A) Use the serializer's or deserializer"s power-up time specifications.

**Q) What is the CC behavior in the case of changing CDS at runtime?**

A) Changing CDS at runtime is not recommended. However, it is possible to change it. Changing CDS at runtime does not disable CC, but it can be used to change the interface between UART and I$^2$C. (Note: The GMSL chip is the I$^2$C master.)

## Configuration

**Q) Can I reconfigure the bit order (such as LSB to MSB vs. MSB to LSB) at the deserializer?**

A) The bit orders are not internally configurable, but you can send video MSB from the LSB slot or vice versa. They can also tie MSB out of the deserializer to LSB on the display side.

**Q) Is it possible to directly connect the MAX9259 with a YUV422 image format? Or do you need a YUV-RGB data converter to input to a Maxim serializer?**

A) The GMSL device accepts RGB or YUV formats as digital inputs and serializes them.

**Q) My MAX9259/MAX9260 chipset is set with active-low AUTOS = low (SEREN = high and CLINKEN = low) in a camera application. After power-up, the camera provides a pixel clock, and the forward channel between the MAX9259 and the MAX9260 is active. I tried to write register 4 (0x04) to change SEREN = low and CLINKEN = high. I can only read the values of register 4, but I cannot write to register 4. There is no problem reading and writing to all other registers. Should I change active-low AUTOS = high to enter Configuration Link Mode at startup? Why can I read register 4 but I cannot write to it?**

A) When SEREN = 0 and CLINKEN = 1 are written, the serial link will be interrupted for some time. During that time, no read or write is possible. The control link (reverse channel only) is not operational for 350µs after stopping/starting the serial link. Additional time may be needed for the forward channel when the forward link is interrupted. When switching from the video link to the configuration link, there will be a change in frequency, and so all running PLLs must relock to a new frequency. This may take longer than 350µs, but its limit will be always less than the "lock time" specification.

**Q) In my setup, the control-direction selection (CDS) pin on the serializer is set to low and on the deserializer set to high, which means that each GMSL chip can connect to its respective µC. If the deserializer's register is set to INTTYPE = 01, then both sides of the µC are accessible to the SerDes register by the UART communication. Under this condition, will I need to provide a solution of my own to handle the bus contention-handling scheme? Does a bus fight exist between the µCs accessing the SerDes register?**

A) Yes. Both µCs are on the same bus, so they need some means to prevent (or correct) bus collision. When needed, the FWDCCEN and REVCCEN bits can be used to disconnect the control channel from the remote side. (See tutorial 5046, "Using GMSL SerDes Devices in a Dual Automotive Electronic Control Unit (ECU) Application" for details.)

**Q) Does MS = L, CSD = L on the serializer and CDS = H on the deserializer make INTTYPE on both the serializer and the deserializer have a "01" setting?**

A) With these settings, both sides have a UART interface. However, the INTTYPE setting is not updated automatically, and the INTTYPE setting is not used (i.e., it is ignored) because CDS already forces the interface type to be UART. The INTTYPE setting of the MAX9263 is effective when CDS = H, and the INTTYPE setting of MAX9264 is effective when CDS = L.

**Q) CDS = L and MS = L enable base mode by a UART. Is the INTTYPE setting of the serializer automatically set to "01" by this pin setting?**

A) No, the INTTYPE setting does not automatically change. The INTTYPE setting of the MAX9263 is effective when CDS = H, and the INTTYPE setting of the MAX9264 is effective when CDS = L. INTTYPE is used to select the control channel interface when that SerDes device is on the peripheral side of the link (not on the µC side).

**Q) What happens if spread-spectrum clocking is turned on at both sides? Is it possible that the buffer will be either over- or underflow?**

A) When the spread spectrum is turned on at the serializer side, the deserializer output also has approximately the same spread percentage and profile. (The CDR tracks it quite well because its bandwidth is significantly higher than the spread-spectrum FM rate.) In this case, the deserializer output would have a 0.5% spread. There are several reasons for this:

The spread added in the deserializer is additive and with respect to the recovered clock. This means that if the serializer and deserializer have exactly the same spread profiles, since their modulation profiles will not be synched, turning the spread spectrum on at both sides might add or subtract one side from the other. If spread is completely synched, almost twice as much spread will be gained; if spread profiles have 180 degrees of phase difference (of the modulation frequency), they might even cancel each other out. That means there is no guaranteed spread profile at the output of the deserializer when spread is turned on at both sides.

If spread profiles are synched and added together, downstream blocks may not be able to handle the new profile due to larger amplitudes (and hence faster variations of frequency for the same frequency modulation). In the case of the MAX9266 and the MAX9268, since 7xPLL occurs after SSPLL, such problems may be encountered.

| Related Parts | | |
|---|---|---|
| MAX9249 | Gigabit Multimedia Serial Link Serializer with LVDS System Interface | Free Samples |
| MAX9257 | Programmable Serializer/Deserializer with UART/I²C Control Channel | Free Samples |
| MAX9258A | Programmable Serializer/Deserializer with UART/I²C Control Channel | 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 | |
| MAX9271 | 16-Bit GMSL Serializer with Coax or STP Cable Drive | Free Samples |
| MAX9272 | 28-Bit GMSL Deserializer for Coax or STP Cable | Free Samples |
| MAX9273 | 22-Bit GMSL Serializer with Coax or STP Cable Drive | 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 5535: http://www.maximintegrated.com/an5535
APPLICATION NOTE 5535, AN5535, AN 5535, APP5535, Appnote5535, Appnote 5535