Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

XR21B1420

Enhanced 1-Ch Full-Speed USB UART

General Description

The XR21B1420 is an enhanced Universal Asynchronous Receiver and Transmitter (UART) bridge to USB interface. The USB interface is fully compliant to the USB 2.0 (Full-Speed) specification with 12 Mbps USB data transfer rate. The USB interface also supports USB suspend, resume and remote wakeup operations. The USB Vendor ID, Product ID, power mode, remote wakeup support, maximum power, and numerous other settings may be programmed in the on-chip OTP memory via the USB interface.

The XR21B1420 includes an internal oscillator and does not require an external crystal/oscillator. Any UART baud rate up to 12 Mbps may be generated with this internal clock and the fractional baud rate generator.

The UART pins may also be configured as GPIO; direction, state, output driver type and input pull-up or pull-down resistors are programmed either through on chip OTP, or on the fly via memory mapped registers.

Large 512-byte TX and RX FIFOs prevent buffer overflow errors and optimize data throughput. Automatic half-duplex direction control and optional multi drop (9-bit) mode simplify both hardware and software in half-duplex RS-485 applications.

The XR21B1420 uses the native OS CDC-ACM driver or an Exar supplied custom driver. Exar provides WHQL/HCK-certified software drivers for Windows 2000, XP, Vista, 7, 8, 8.1 as well as software drivers for Windows CE, Linux and Mac OS X. Full source code is available.

The XR21B1420 operates from a single 5V or 3.3V power supply. When powered with 5V input, a regulated 3.3V output is supplied.

FEATURES

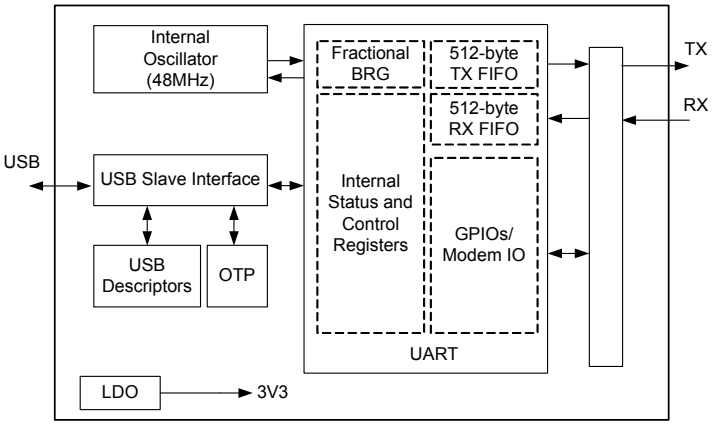
- ±15kV ESD on USBD+/USBD-
- USB 2.0 Compliant, Full-Speed (12Mbps)
- Unique pre-programmed USB serial number
- Internally generated 48MHz core clock
- Enhanced UART features
  - Baud rates up to 12 Mbps
  - Fractional Baud Rate Generator
  - 512-byte TX and 512-byte RX FIFOs
  - Auto Hardware / Software Flow Control
  - Multidrop and Half-Duplex Modes
  - Auto RS-485 Half-Duplex Control
  - Selectable GPIO or Modem I/O
- Up to 10 GPIOs
- 5V tolerant GPIO inputs
- Suspend state GPIO configuration
- Configurable clock output
- 28-pin QFN package
- Industrial -40°C to +85°C Temperature Range

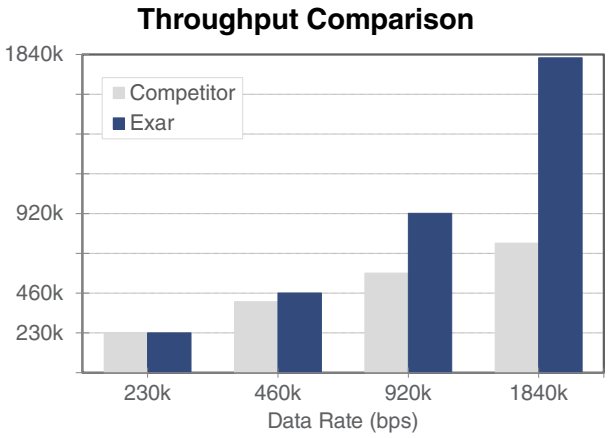
APPLICATIONS

- Building Automation
- Security Systems
- Factory and Process Control
- ATM Terminals
- USB to Serial Controllers

Ordering Information – page 60

Block Diagram





© 2014 Exar Corporation

Absolute Maximum Ratings

Stresses beyond the limits listed below may cause permanent damage to the device. Exposure to any Absolute Maximum Rating condition for extended periods may affect device reliability and lifetime.

Supply Voltage (VCC_REG).....+5.75V

Supply Voltage (VCC).....+4V

Input Voltage (VBUS_SENSE).....-0.3 to +5.75V

Input Voltage (All other pins).....-0.3 to +5.6V

Junction Temperature.....125°C

Operating Conditions

Operating Temperature Range.....-40°C to +85°C

Electrical Characteristics

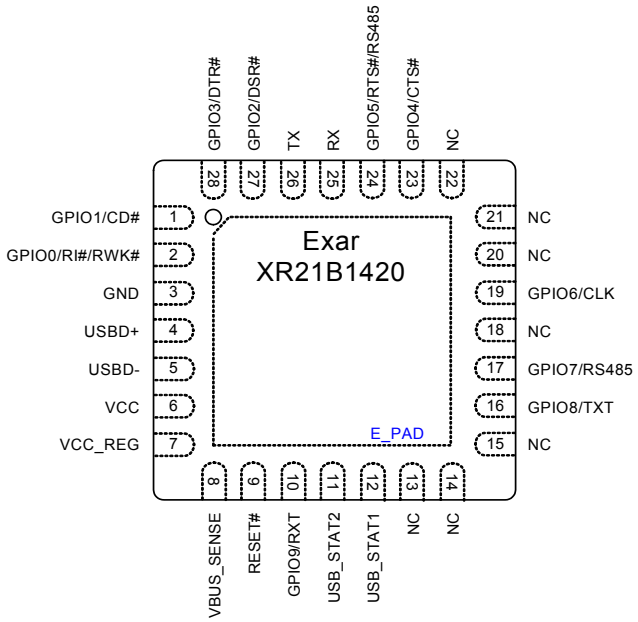
Unless otherwise noted: TA = -40°C to +85°C, VCC_REG = +4.4V to +5.25V or +3.0V to +3.6V.

Symbol	Parameter	Conditions		Min	Typ	Max	Units
Power							
I_{CC}	Power Supply Current	VCC_REG = +4.4V to +5.25V			15	23	mA
I_{SUSP}	Lowpower Mode Current				0.85	1.35	mA
V_{OUT}	Regulated Output Voltage (VCC pin)	VCC_REG = +4.4V to +5.25V. Maximum output current = 200 mA including the supply current of the XR21B1420.		3	3.3	3.6	V
UART, USB_STAT and GPIO Pins							
V_{IL}	Input Low Voltage			-0.3		0.25* VCC	V
V_{IH}	Input High Voltage			0.70* VCC		5.5	V
V_{OL}	Output Low Voltage	IOL = 1mA, VCC = +3.6V				0.5	V
V_{OH}	Output High Voltage	IOH = -1.5mA, VCC = +3.6V		2.8		VCC	V
I_{IL}	Input Low Leakage Current	VCC = +3V to +3.6V, VCC_REG = +4.4V to +5.25V, V_{INPUT} = 0V				±10	µA
I_{IH}	Input High Leakage Current	VCC = +3V to +3.6V, VCC_REG = +4.4V to +5.25V, V_{INPUT} = +3.3V				±10	µA
		VCC = +3V to +3.6V, VCC_REG = +4.4V to +5.25V, V_{INPUT} = +5.5V				±120	µA
C_{IN}	Input Pin Capacitance					5	pF

© 2014 Exar Corporation

Symbol	Parameter	Conditions		Min	Typ	Max	Units
USB I/O Pins							
V_{IL}	Input Low Voltage			-0.3		0.8	V
V_{IH}	Input High Voltage			2.0		5.5	V
V_{OL}	Output Low Voltage	External 15kΩ to GND on USBD+ and USBD- pins		0		0.3	V
V_{OH}	Output High Voltage	External 15kΩ to GND on USBD+ and USBD- pins		2.8		3.6	V
V_{DrvZ}	Driver Output Impedance			28		44	Ω

Pin Configuration



Top View

Pin Assignments

Pin No.	Pin Name	Type	Description
1	GPIO1/CD#	I/O	General purpose I/O, or UART Carrier-Detect input (active low). Defaults to GPIO input with internal pull-up resistor.
2	GPIO0/RI#/RWK#	I/O	General purpose I/O, or UART Ring-Indicator input (active low), or Remote Wakeup input. Defaults to GPIO input with internal pull-up resistor. See “Remote Wakeup” on page 7.
3	GND	PWR	Power supply common, ground.
4	USBD+	I/O	USB port differential data positive. This pin has internal pull-up resistor compliant to the USB 2.0 specification. The ESD protection on this pin is ±15kV HBM.
5	USBD-	I/O	USB port differential data negative. The ESD protection on this pin is ±15kV HBM.
6	VCC	PWR	3.3V power to the device, or 3.3V power output from the device when 5V power is supplied to VCC_REG pin. 3.3V output power can source up to 200 mA maximum (including the device) and should be decoupled by minimum of 4.7µF ceramic capacitor. See “USB Power Modes” on page 9.
7	VCC_REG	PWR	5V or 3.3V power to the device. In bus-powered mode, connect VBUS power from the USB host to this pin and to the VBUS_SENSE pin. See Figure 1. In self-powered mode, connect on-board 5V or 3.3V source to this pin and VBUS from the USB host to the VBUS_SENSE pin. See Figure 2 and Figure 3.
8	VBUS_SENSE	I	Must be connected to VBUS power from the USB host PC. This pin is used to disable the internal pull-up resistor on the USBD+ signal when VBUS is not present.
9	RESET#	I/O OD	Active low open drain output. Asserted at power on or any time device is reset by either register or USB bus reset. As an input, must be asserted for at least 15µs to force a device reset. Reset pulse width input of shorter than 15µs will have unknown effects. A weak internal pull-up resistor provides noise immunity if left unconnected.
10	GPIO9/RXT	I/O	General purpose I/O, or UART receive data indicator. Defaults to GPIO input with internal pull-up resistor. See “TXT and RXT Pins” on page 15. When configured as receive indicator, this pin will toggle at ~10Hz intervals while the UART is receiving data.
11	USB_STAT2	O / OD	This pin has the same functionality as the USB_STAT1 pin. However, the default output for this pin is active low polarity, asserted whenever the XR21B1420 is placed into a suspended state. This default may be changed via the PIN_CFG_USB_STAT2 register.
12	USB_STAT1	O	The USB_STAT1 output pin may be used to indicate any of three USB status conditions: 1. USB_STAT1 is asserted when the USB host asserts USB reset. 2. USB_STAT1 is asserted when the USB host PC places the XR21B1420 device into the suspend state. 3. USB_STAT1 is asserted when it is not safe to draw the amount of current requested in the Device Maximum Power field of the Configuration Descriptor. a. For a low power device (<=1 unit load or 100mA, bMaxPower <= 0x32), USB_STAT1 will be asserted when the USB UART is in the suspend mode or when it is not yet configured. b. For a high power device (bMaxPower > 0x32), USB_STAT1 will be asserted when the USB UART is in the suspend mode or when it is not yet configured. The assertion polarity and status condition are selectable via the PIN_CFG_STAT1 register. The USB_STAT pin will be de-asserted whenever the selected condition(s) is/are not met. The default output for this pin is active high polarity, asserted whenever the XR21B1420 is placed into a suspended state.
13	NC	-	No Connect.
14	NC	-	No Connect.
15	NC	-	No Connect.

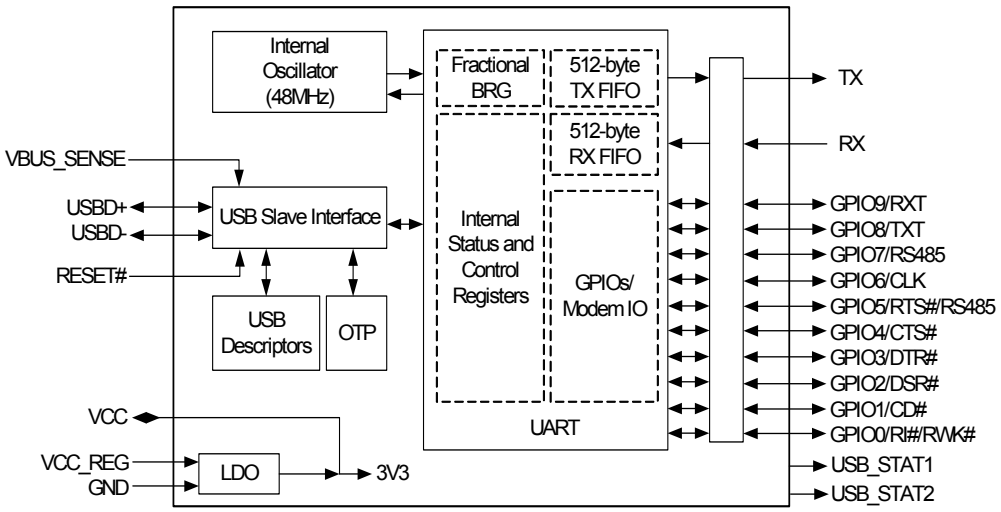
© 2014 Exar Corporation

Pin No.	Pin Name	Type	Description
16	GPIO8/TXT	I/O	General purpose I/O, or UART transmit data indicator. Defaults to GPIO input with internal pull-up resistor. See "TXT and RXT Pins" on page 15. When configured as transmit indicator, this pin will toggle at ~10Hz intervals while the UART is transmitting data.
17	GPIO7/RS485	I/O	General purpose I/O, or RS-485 half-duplex enable output. Defaults to GPIO input with internal pull-up resistor.
18	NC	-	No Connect.
19	GPIO6/CLK	I/O	General purpose I/O, or clock or pulse output. Defaults to GPIO input with internal pull-up resistor. See "Programmable Output Clock" on page 13.
20	NC	-	No Connect.
21	NC	-	No Connect.
22	NC	-	No Connect.
23	GPIO4/CTS#	I/O	General purpose I/O, or UART Clear-to-Send input (active low). Defaults to GPIO input with internal pull-up resistor. See "Automatic RTS/CTS Hardware Flow Control" on page 13.
24	GPIO5/RTS#/RS485	I/O	General purpose I/O, or UART Request-to-Send output (active low), or auto RS-485 half-duplex control. Defaults to GPIO input with internal pull-up resistor. See "Automatic RTS/CTS Hardware Flow Control" on page 13 or "Multidrop mode with address matching" on page 14.
25	RX	I	UART Receive Data.
26	TX	O	UART Transmit Data.
27	GPIO2/DSR#	I/O	General purpose I/O, or UART Data-Set-Ready input (active low). Defaults to GPIO input with internal pull-up resistor. See "Automatic DTR/DSR Hardware Flow Control" on page 14.
28	GPIO3/DTR#	I/O	General purpose I/O, or UART Data-Terminal-Ready push-pull output (active low). Defaults to GPIO input with internal pull-up resistor. See "Automatic DTR/DSR Hardware Flow Control" on page 14.

Type: I = Input, O = Output, I/O = Input/Output, PWR = Power, OD = Open-Drain

© 2014 Exar Corporation

Functional Block Diagram



Functional Description

USB Interface

The USB interface of the XR21B1420 is compliant with the USB 2.0 Full-Speed Specifications.

The XR21B1420 uses the following set of parameters:

- 1 Control Endpoint
  - Endpoint 0 as outlined in the USB specifications
- 1 Configuration is supported
- 1 Interface for the UART
  - Bulk-in and bulk-out endpoints
  - Interrupt-in endpoint for notifications

USB Vendor and Product IDs

Exar's USB Vendor ID is 0x04E2. This is the default Vendor ID that is used for the XR21B1420. Customers may obtain their own Vendor ID from USB.org. The default USB Product ID for the XR21B1420 is 0x1420. Upon request, Exar will provide up to 8 PID values for use with Exar's VID. The VID and PID can be changed using the VID and PID fields. Refer to Table 1

USB Suspend

All USB peripheral devices must support the USB suspend mode. Per USB standard, the XR21B1420 device will begin to enter the suspend state if it does not detect any activity, (including Start of Frame or SOF packets) on its USB data lines for 3 ms. The peripheral device must then reduce power consumption from VBUS power within the next 7 ms to the allowed limit of 2.5 mA for the suspended state. Note that in this context, the "device" is all circuitry (including the XR21B1420) that draws power from the host VBUS.

Remote Wakeup

If the XR21B1420 device has been placed into the suspend state by the USB host, a high to low transition on the RI#/RWK# pins can be used to request that the host exit the suspended state. By default the XR21B1420 device reports in its USB device attributes that it supports remote wakeup. The RI#/RWK# pin of the UART is enabled for remote wakeup signaling if the default configuration as an input pin has not been changed. Additionally, the RX pin of the UART may also be enabled via OTP to support remote wakeup. Note that the CDC driver does not support remote wakeup.

USB Strings

USB specifies three character string descriptors that are provided to the USB host during enumeration in string descriptors: the manufacturer, product and serial strings. The default manufacturer and product strings for the XR21B1420 device are "Exar Corp." and, "Exar USB UART", respectively. The serial number string is a unique alpha-numeric string programmed into the device at the factory. All character strings use Unicode UTF-16LE format by default, but the Unicode language ID may be changed for the manufacturer and product strings. The default character string language ID is US English. If the language ID is modified via OTP, the serial number string should also be modified accordingly. To ensure unique serial number strings, it is recommended that the factory pre-programmed serial number string be used.

© 2014 Exar Corporation

Table 1: USB String Descriptor Defaults

Descriptor	Value
Exar USB Vendor ID	0x04E2
Exar USB Product ID	0x1420
Manufacturer String	Exar Corp.
Product String	Exar USB UART

Device Driver

The XR21B1420 device may be used with either a standard CDC-ACM driver or an Exar supplied custom driver. The CDC-ACM driver is native to the Operating System. In Linux, the CDC-ACM driver will automatically load for the XR21B1420, but in the Windows OS, an extra INF file is required to install the CDC-ACM driver. The custom drivers must also be installed, although for Windows 7 OS and newer with Internet access and Windows updates set to automatic, the latest Windows-Certified (WHQL/HCK) driver will be downloaded and installed automatically.

CDC-ACM Driver

Because the CDC-ACM driver has no ability to access the XR21B1420 internal device registers, the device is initialized to certain hardware defaults. By default the XR21B1420 enables hardware RTS/CTS flow control, GPIO7 is set as active high auto RS-485 half-duplex control, and RI, CD and DSR pins are enabled to be interrupt sensitive. These settings are listed in Table 2. Additionally, the low latency threshold in CDC mode is automatically set to 40,960 bps. Refer to “RX FIFO Low Latency” on page 12. This threshold may be modified in the OTP CDC_ACM_BAUD_THRESH locations.

Table 2: XR21B1420 Register Defaults with CDC-ACM Driver

Register	Value	Notes
FLOW_CONTROL	0x0001	Hardware flow control
GPIO_MODE	0x0339	RTS / CTS flow control, GPIO7 is used as RS-485 half-duplex enable (RS485) with active high polarity. GPIO6 is a GPIO input, RXT and TXT remain enabled.
GPIO_DIRECTION	0x0028	DTR / RTS are configured as outputs (TXT, RXT, CLK and RS485 are also special function outputs). All other GPIOs are configured as inputs.
GPIO_INT_MASK	0x03F0	RI, CD and DSR are interrupt sensitive, i.e. can cause a USB interrupt to be generated.

Custom Exar Driver

Custom Windows and Linux drivers are available from Exar. The custom driver allows software applications to make full use of the XR21B1420 register set and features.

Note that a custom driver must always immediately set CUSTOM_DRIVER bit-0 = 1. Once CUSTOM_DRIVER bit-0 is set, the custom driver can use standard CDC-ACM commands without the XR21B1420 automatically changing to the settings in the Table 2.

© 2014 Exar Corporation

USB Power Modes

The XR21B1420 device may be configured in any of the following power modes: bus-powered, self-powered 5V, or self-powered 3.3V. In all three modes, the VBUS power signal from the USB host must be connected to the VBUS_SENSE pin of the device.

The default power mode for the XR21B1420 is bus powered. In this mode, the USB device's maximum power requirement from the host must be specified. In this context, the USB device includes all components on the PCB that will draw power from the USB host VBUS power. The default maximum power for the XR21B1420 is 100mA. This may be changed using the Attributes field in the OTP.

Bus-Powered

In bus-powered mode, VBUS from the USB cable supplies 5V to the XR21B1420 device. The VCC pin will supply a 3.3V output.

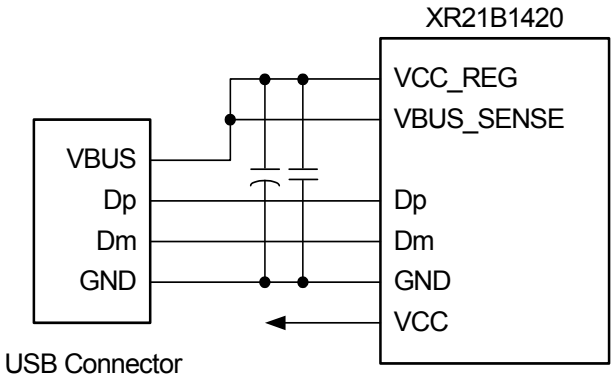


Figure 1: Bus-Powered Mode

Self-Powered 5V

In self-powered 5V mode, a local source provides 5V to the XR21B1420 device. The USB attributes should be changed in the OTP to correctly report self-powered mode. The VCC pin will supply a 3.3V output.

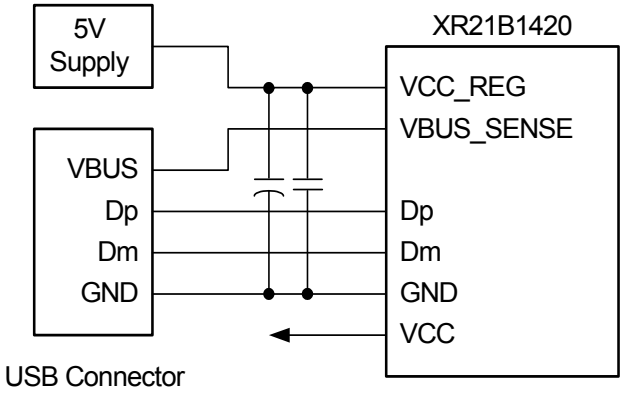


Figure 2: Self-Powered 5V Mode

Self-Powered 3.3V

In self-powered 3.3V mode, a local source provides 3.3V to both the VCC_REG and VCC pins of the XR21B1420 device. The USB attributes should be changed in the OTP to correctly report self-powered mode.

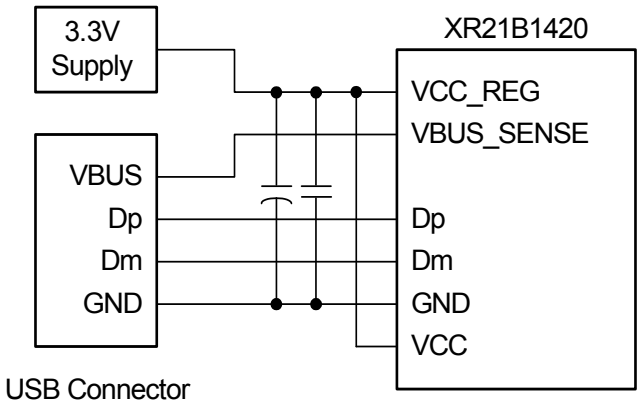


Figure 3: Self-Powered 3.3V Mode

Reset

The XR21B1420 has three different types of resets: power-on reset or POR, hardware reset, and USB bus reset. The results of each of the three types of resets are listed in Table 3.

Table 3: Device Resets

Reset Type	Device Actions
Power On Reset (POR)	Resets all registers and pins to default states including any OTP modifications. Locks OTP from further writes if Global Lock is set.
Hardware Reset	Resets all registers and pins to default states including any OTP modifications. Locks OTP from further writes if Global Lock is set.
USB Bus Reset	Resets USB Interface, re-enumerate device, reset all internal states, clear UART FIFOs. Does not reset registers or pin configurations.

UART

The UART may be configured via USB control transfers from the USB host. The UART transmitter and receiver sections are described separately in the following sections. At power-up, the XR21B1420 will default to 115.2 kbps, 8 data bits, no parity bit, 1 stop bit, and no flow control. If a standard CDC driver accesses the XR21B1420, these defaults will be changed. See “Device Driver” on page 8.

Transmitter

The transmitter consists of a 512-byte TX FIFO and a Transmit Shift Register (TSR). Once a Set transmit data interrupt out or bulk-out packet has been received and the CRC has been validated, the data bytes in that packet are written into the TX FIFO. Data from the TX FIFO is transferred to the TSR when the TSR is idle or has completed sending the previous data byte. The TSR shifts the data out onto the TX output pin at the selected baud rate. The transmitter sends the start bit followed by the data bits (starting with the LSB), inserts the proper parity-bit if enabled, and adds the stop-bit(s). The transmitter may be configured for 5, 6, 7 or 8 data bits with or without parity or 9 data bits without parity. If 5, 6, 7 or 8 bit data with

parity is selected, the TX FIFO contains 8 bits data and the parity bit is automatically generated and transmitted. If 9 bit data is selected, parity cannot be generated. The 9th bit will not be transmitted unless the wide mode is enabled.

Wide Mode Transmit

When both 9 bit data and wide mode are enabled, two bytes of data will be written into the TX FIFO. The first byte is the first 8 bits (data bits 7-0) of the 9-bit data. Bit-0 of the second byte is bit-8 of the 9-bit data. The data that is transmitted on the TX pin is as follows: start bit, 9-bit data, stop bit. Wide mode may be enabled using the TX_WIDE_MODE and RX_WIDE_-MODE registers.

Receiver

The receiver consists of a 512-byte RX FIFO and a Receive Shift Register (RSR). Data that is received in the RSR via the RX pin is transferred into the RX FIFO. Data from the RX FIFO is sent to the USB host by in response to a bulk-in request. Depending on the mode, error / status information for that data character may or may not be stored in the RX FIFO with the data.

Normal receive operation with 5, 6, 7 or 8-bit data

Received data is stored in the RX FIFO. Any parity, framing or overrun error or break status information related to the data is discarded. The receive data format is shown in Figure 4.

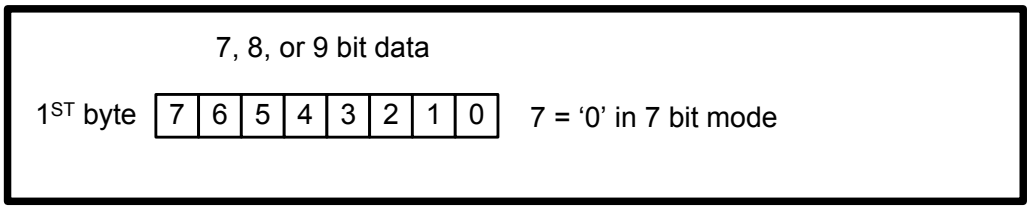


Figure 4: Receive Data Format

Normal receive operation with 9-bit data

The first 8 bits of data received is stored in the RX FIFO. The 9th bit as well as any parity, framing or overrun error or break status information related to the data is discarded.

Wide mode receive operation with 5, 6, 7 or 8-bit data

Two bytes of data are loaded into the RX FIFO for each byte of data received. The first byte is the received data. The second byte consists of the error bits and break status. Wide mode receive data format is shown in Figure 5.

© 2014 Exar Corporation

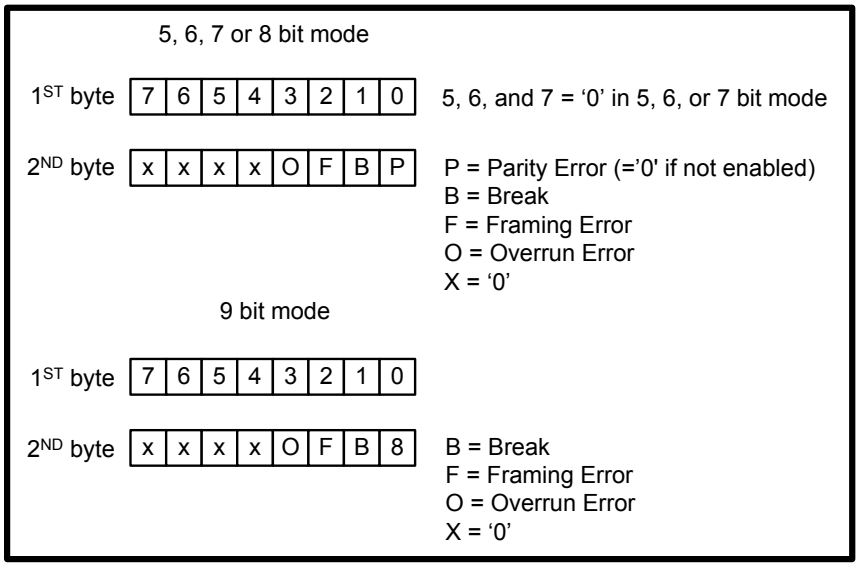


Figure 5: Wide Mode Receive Data Format

Wide mode receive operation with 9-bit data

Two bytes of data are loaded into the RX FIFO for each byte of data received. The first byte is the first 8 bits of the received data. The 9th bit received is stored in the bit 0 of the second byte. The parity bit is not received / checked. The remainder of the 2nd byte consists of the framing and overrun error bits and break status.

Error flags are also available from the ERROR_STATUS register and the interrupt packet, however these flags are historical flags indicating that an error has occurred since the previous request. Therefore, no conclusion can be drawn as to which specific byte(s) may have contained an actual error.

RX FIFO Low Latency

In normal operation all bulk-in transfers will be of maxPacketSize (64) bytes to improve throughput and to minimize host processing. When there are 64 bytes of data in the RX FIFO, the XR21B1420 will acknowledge a bulk-in request from the host and transfer the data packet. If there are less than 64 bytes in the RX FIFO, the XR21B1420 may respond to the bulk-in request with a NAK indicating that data is not ready to transfer at that time. However, if there are less than 64 bytes in the RX FIFO and no data has been received for more than 3 character times, the XR21B1420 will acknowledge the bulk-in request and transfer any data in the RX FIFO to the USB host.

In some cases, especially when the baud rate is low, this behavior may increase latency unacceptably. The XR21B1420 has a low latency register bit that will enable the XR21B1420 to immediately transfer any received data in the RX FIFO to the USB host without waiting for 3 character times. The custom driver may be used to automatically set the RX_FIFO_LOW_LATENCY register to enable low latency mode, or the user may manually set it. With the CDC-ACM driver, the low latency mode is automatically set whenever the baud rate is set to a value of less than 40960 bps using the CDC_ACM_IF_SET_LINE_CODING command.

GPIO

Each UART has 10 GPIO pins in addition to the TX and RX pins. Each GPIO pin may also be configured for one or more special functions. All GPIO pins as well as USB_STAT1 and USB_STAT2 may be configured for a variety of pin type options using the GPIO_MODE register or by writing the OTP using XR_SET_OTP. All enabled pull-up and pull-down resistors are maintained during the USB suspend state. Pin configurations set using XR_SET_OTP are enabled following the next power-up reset and are permanent. During USB bus reset, resistors are disabled and are re-enabled after bus reset is de-asserted. Pin configurations set using the GPIO_MODE register will be lost after POR or USB bus reset.

© 2014 Exar Corporation

Programmable Output Clock

The GPIO6/CLK pin may be enabled as a clock output using the GPIO_MODE register. The OUTCLK register can be used to program the output frequency of the clock from 24 MHz down to approximately 47 KHz. The duty cycle can also be programmed from 50/50 to a single low or high going pulse. The default values of zero for both DIV_HI and DIV_LO in the OUTCLK register will result in a frequency of 24 MHz. For any non-zero values for DIV_HI and DIV_LO, the clock frequency is determined by the formula:

FREQ = 24 MHz / (DIV_HI + DIV_LO). The duty cycle is determined by the ratio of DIV_HI to DIV_LO.

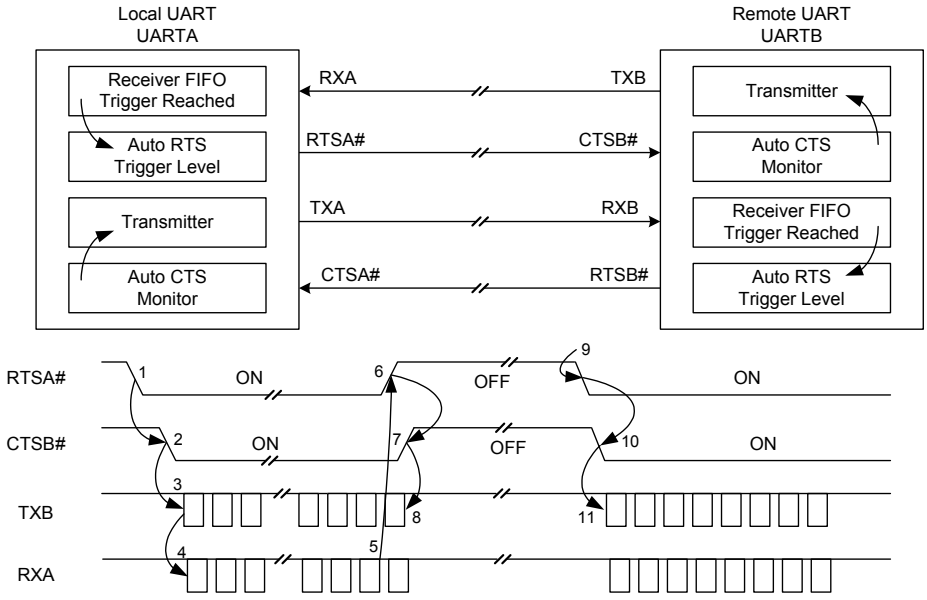
Flow Control

The XR21B1420 is able to perform both hardware and software flow control. Both hardware and software flow control modes are configured via the GPIO_MODE and FLOW_CONTROL registers. In both modes, flow control is asserted when the bytes in the RX FIFO reach the watermark set in the RX_THRESHOLD register.

Hardware flow control can either be RTS/CTS or DTR/DSR controlled. Note that although the default pin configuration for GPIO5/RTS#/RS485 and GPIO4/CTS# are for RTS output and CTS input respectively, the hardware RTS/CTS flow control mode must be set in the FLOW_CONTROL register in order to utilize the flow control functionality.

Automatic RTS/CTS Hardware Flow Control

Automatic RTS flow control is used to prevent data overrun errors in the local RX FIFO using the RTS signal to the remote UART. The RTS signal will be asserted (low) when there are less than 450 bytes in the receive FIFO. When the RX FIFO reaches the 450 byte threshold, the RTS pin will be de-asserted. The CTS# input is monitored by the remote UART to suspend/restart the local transmitter. Refer to Figure 6. Conversely, when the remote UART reaches its receive FIFO threshold, its RTS will be de-asserted, and the XR21B1420 CTS input will cause the device to suspend data transmission.



1) COM port opened, RX FIFO empty, RTSA# output is asserted
2) Signal propagated to CTSB# input
3) Data bytes enter TX FIFO, begin transmitting on TXB
4) Data propagates to Receiving device RXA
5) RX FIFO reaches threshold
6) RTSA# de-asserts
7) Signal propagates to CTSB# input
8) Transmission stops on TXB
9) USB Bulk-In empties RX FIFO below threshold, RTSA# is asserted
10) Signal propagated to CTSB# input
11) Data bytes resume transmitting on TXB

© 2014 Exar Corporation

Figure 6: Auto RTS and CTS Flow Control Operation

Automatic DTR/DSR Hardware Flow Control

Auto DTR/DSR hardware flow control behaves the same as the Auto RTS/CTS hardware flow control described above except that it uses the DTR# and DSR# signals. GPIO2 and GPIO3 become DSR# and DTR#, respectively, when the GPIO_MODE register is configured for DTR/DSR hardware flow control.

Automatic XON/XOFF Software Flow Control

When software flow control is enabled, the XR21B1420 compares the receive data characters with the programmed XON or XOFF characters. If the received character matches the programmed XOFF character, the XR21B1420 will halt transmission as soon as the current character has completed transmission. Data transmission is resumed when a received character matches the XON character.

In the receive data direction, the XOFF character will be sent when there are 450 bytes in the receive FIFO. When there are again less than 450 bytes in the RX FIFO, the XON character will be sent. This threshold may be changed using the RX_THRESHOLD register.

Software flow control is enabled / disabled by the FLOW_CONTROL register. Additionally, the XON_CHAR and XOFF_CHAR registers may be used to configure the start (XON) and stop (XOFF) characters.

Multidrop mode with address matching

The XR21B1420 device has two address matching modes which are set by the FLOW_CONTROL and GPIO_MODE registers. These modes are intended for use in a multi-drop network application. Address matching may be used with any size data character, as well as with and without parity. An address match occurs when the last (most significant) received data bit or the parity bit, if there is one, is a ’1’ and the address matches the value stored in either the XON_CHAR or XOFF_CHAR register. To send an address byte use 5, 6, 7, 8 or 9 bit data with either the most significant data bit a ’1’ or if parity is used, set mark parity. To send data bytes, the most significant data bit must be a ’0’ or use space parity.

Receiver

If an address match occurs in either of the address matching modes, the address byte and all subsequent data bytes will be loaded into the RX FIFO. The UART Receiver will automatically be disabled when an address byte is received that does not match the values in the XON_CHAR or XOFF_CHAR characters.

Transmitter

In flow control mode 3, the UART transmitter will transmit irrespective of the RX address match. In flow control mode 4, the UART will only transmit following an RX address match.

Programmable Turn-Around Delay

By default, the selected RS-485 half-duplex enable pin (either GPIO7/RS485 or GPIO5/RTS#/RS485) will be de-asserted immediately after the stop bit of the last byte has been shifted. However, this may not be ideal for systems where the signal needs to propagate over long cables. Therefore, the de-assertion of the RS-485 half-duplex enable can be delayed from 1 to 15 bit times via the XCVR_EN_DELAY register to allow for the data to reach distant UARTs.

UART Half-Duplex Mode

In UART half-duplex mode, the UART will ignore any data on the RX input when the UART is transmitting data. The half-duplex mode can be configured using the FLOW_CONTROL register.

© 2014 Exar Corporation

IR Mode

The XR21B1420 supports IR mode at a maximum baud rate of 2.5 Mbaud with transmit pulses of 3/16th or 4/16th of a bit period and centered in the bit period. Receive data may be inverted to conform to some manufacturer's non-standard devices. IR mode is disabled by default but may be enabled by the IR_MODE register.

USB_STAT Pins

The XR21B1420 has two USB_STAT output pins that may be used to indicate 3 different statuses in either positive or negative polarity. The SUSPEND status indicates that the XR21B1420 device has been placed into a suspended state by the USB host. This output can then be used by external circuitry, for example, to power down devices in order to meet USB requirements for suspend mode. The LOW_POWER status is similar to the SUSPEND status, but LOW_POWER is also asserted for high power devices (any device that consumes more than 100 mA of VBUS power from the USB host), before the device is configured during enumeration by the USB host. For low power devices (devices that consume 100 mA or less of VBUS power), SUSPEND and LOW_POWER status outputs are functionally the same. Lastly, the BUS_RESET output status is asserted any time the XR21B1420 device is being reset by the USB host. This status output could be used, for example, by an FPGA or other logic device to synchronize this external logic with the XR21B1420 device.

Suspend Mode Settings

The USE_SUSPEND bit controls the GPIO pins when the XR21B1420 device is suspended by the USB host. If USE_SUSPEND is cleared to '0', the GPIO pins retain their output states when the device is suspended. When USE_SUSPEND is set to '1', the GPIO pin's behavior is defined by the SUSPEND_STATE and SUSPEND_MODE registers, with the following exceptions: GPIO0/CLK when configured as an output clock will always be driven low, i.e the clock output will stop, and GPIO1/RTS#/RS485 or GPIO3/RS485 when configured as auto. RS-485 half-duplex enable will always be de-asserted. Note that USE_SUSPEND does not affect the UART RX and TX pins. During suspend state, RX and TX will always idle to a logic '1' state.

The SUSPEND_STATE field will set or clear the GPIO pins and the SUSPEND_MODE field will configure GPIO outputs as either open drain or push-pull outputs. SUSPEND_STATE and SUSPEND_MODE may be configured through registers or OTP. As opposed to OTP configuration, register configurations are not retained if the power is lost or the bus is reset.

TXT and RXT Pins

The Transmit toggle and Receive toggle pins "toggle" at a rate of approximately 10 Hz whenever the UART transmit and receive pins (respectively) are active.

OTP

The OTP is an on-chip non-volatile memory, that is incrementally one-time programmable via the USB interface. Some bits are pre-programmed at the factory and caution must be taken not to program any locations except those user defined addresses given in this data sheet. Once a specific portion of the OTP is programmed, the PROG bit for that section of the OTP must be set and further changes to that section will not be allowed.

© 2014 Exar Corporation

USB Control Commands

The following table shows all of the USB Control Commands that are supported by the XR21B1420. Commands include standard USB commands, CDC-ACM commands and Exar vendor specific commands. The device internal registers are accessed using the vendor specific XR_GET_REG and XR_SET_REG, XR_GET_REVISION, XR_GET_USB_STAT and XR_SET_USB_STAT vendor specific commands.

Table 4: Supported USB Control Commands

Name	Request Type	Request	Value		Index		Length		Description
			LSB	MSB	LSB	MSB	LSB	MSB	
DEV GET_STATUS	0x80	0x0	0x0	0x0	0x0	0x0	0x2	0x0	Device: remote wake-up + self-powered
IF GET_STATUS	0x81	0x0	0x0	0x0	0x0	0x0	0x2	0x0	Interface: zero
EP GET_STATUS	0x82	0x0	0x0	0x0	0x0,0x4 ,0x84	0x0	0x2	0x0	Endpoint: halted
DEV CLEAR_FEA-TURE	0x00	0x1	0x1	0x0	0x0	0x0	0x0	0x0	Device remote wake-up
EP CLEAR_FEA-TURE	0x02	0x1	0x0	0x0	0x0,0x4 ,0x84	0x0	0x0	0x0	Endpoint halt
DEV SET_FEATURE	0x00	0x3	0x1	0x0	0x0	0x0	0x0	0x0	Device remote wake-up
EP SET_FEATURE	0x02	0x3	0x0	0x0	0x0,0x4 ,0x84	0x0	0x0	0x0	Endpoint halt
SET_ADDRESS	0x00	0x5	addr	0x0	0x0	0x0	0x0	0x0	addr = 1 to 127
GET_DESCRIPTOR	0x80	0x6	0x0	0x1	0x0	0x0	len MSB	len MSB	Device descriptor
GET_DESCRIPTOR	0x80	0x6	0x0	0x2	LangID	LangID	len MSB	len MSB	Configuration descriptor
GET_DESCRIPTOR	0x80	0x6	0x0	0x3	0x0	0x0	len MSB	len MSB	String descriptor
GET_CONFIGURA-TION	0x80	0x8	0x0	0x0	0x0	0x0	0x1	0x0	
SET_CONFIGURA-TION	0x00	0x9	n	0x0	0x0	0x0	0x0	0x0	n = 0, 1
GET_INTERFACE	0x81	0x10	0x0	0x0	0x0	0x0	0x1	0x0	
CDC_ACM_IF SET_LINE_CODING	0x21	0x20	0x0	0x0	0x0	0x0	0x7	0x0	Set the UART baud rate, parity, stop bits, etc.
CDC_ACM_IF GET_LINE_CODING	0xA1	0x21	0x0	0x0	0x0	0x0	0x7	0x0	Get the UART baud rate, parity, stop bits, etc.
CDC_ACM_IF SET_CONTROL_ LINE_STATE	0x21	0x22	0x0	0x0	0x0	0x0	0x7	0x0	Set/Clear DTR in CDC-ACM mode.

© 2014 Exar Corporation

Table 4: Supported USB Control Commands

Name	Request Type	Request	Value		Index		Length		Description
			LSB	MSB	LSB	MSB	LSB	MSB	
CDC_ACM_IF SEND_BREAK	0x21	0x23	val LSB	val MSB	0x0	0x0	0x0	0x0	Send a break for the specified duration.
XR_GET_CHIP_ID	0xC0	0xFF	0x0	0x0	0x0	0x0	0x6	0x0	Get Exar VID (2 bytes), PID (2 bytes) and bcdDevice (2 bytes)
XR_SET_REG See Table 5	0x41	0x0	write-data LSB	write-data MSB	write addr	0x0	0x0	0x0	Vendor specific register access.
XR_GET_REG See Table 5	0xC1	0x0	0x0	0x0	read addr	0x0	0x2	0x0	Vendor specific register access.
XR_GET_REVISION See Table 5	0xC0	0x0	0x0	0x0	0x60	0x02	0x2	0x0	Vendor specific register access.
XR_SET_USB_STAT See Table 5	0x40	0x0	write-data LSB	write-data MSB	0x62	0x02	0x0	0x0	Vendor specific register access.
XR_SET_USB_STAT See Table 5	0xC0	0x0	0x0	0x0	0x62	0x02	0x2	0x0	Vendor specific register access.

© 2014 Exar Corporation

Register Set Description

The internal register set of the XR21B1420 controls the UART functionality, basic functionality of the FIFOs, OTP controls, as well as registers associated with the processing of driver commands. All registers are accessible via the USB interface using the XR_SET_REG and XR_GET_REG USB commands, except for the REVISION_ID and USB_STAT registers which are accessible with the XR_GET_REVISION and XR_GET/SET_USB_STAT commands respectively. Note that the UART_ENABLE register should be used to disable the UART prior to any register write and re-enable the UART following any single or sequence of register writes except for the GPIO_SET, GPIO_CLEAR, TX_BREAK and ERROR_STATUS registers.

All registers are 16 bits wide. The upper byte of single byte registers as well as bit locations with field label of '0' in Table 5 are reserved. All reserved bits must be written as zeroes when modifying register contents.

Table 5: XR21B1420 Register Map

Address	Register Name	Bit 7 (15)	Bit 6 (14)	Bit 5 (13)	Bit 4 (12)	Bit 3 (11)	Bit 2 (10)	Bit 1 (9)	Bit 0 (8)
0x000	UART_ENABLE	0	0	0	0	0	0	RX	TX
0x006	FLOW_CONTROL	0	0	0	0	Half-Duplex	Flow Control Mode Select		
0x007	XON_CHAR	VALUE							
0x008	XOFF_CHAR	VALUE							
0x009	ERROR_STATUS	Break Status	Overrun Error	Parity Error	Framing Error	Break Error	0	0	0
0x00A	TX_BREAK[15:8]	VALUE (MSB)							
	TX_BREAK[7:0]	VALUE (LSB)							
0x00B	XCVR_EN_DELAY	0	0	0	0	Delay			
0x00C	GPIO_MODE[15:8]	0	0	0	0			RXT_EN	TXT_EN
	GPIO_MODE[7:0]	CLK_EN	RS485_SEL		XCVR Enable Pin	XCVR Enable Polarity	Mode Select		
0x00D	GPIO_DIRECTION[15:8]	0	0	0	0	0	0	GPIO9	GPIO8
	GPIO_DIRECTION[7:0]	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x00E	GPIO_SET[15:8]	0	0	0	0	0	0	GPIO9	GPIO8
	GPIO_SET[7:0]	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x00F	GPIO_CLEAR[15:8]	0	0	0	0	0	0	GPIO9	GPIO8
	GPIO_CLEAR[7:0]	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x010	GPIO_STATE[15:8]	0	0	0	0	TX	RX	GPIO9	GPIO8
	GPIO_STATE[7:0]	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x011	GPIO_INT_MASK[15:8]	0	0	0	0	0	RX	GPIO9	GPIO8
	GPIO_INT_MASK[7:0]	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x012	CUSTOMIZED_INT	0	0	0	0	0	0	INT_ BREAK_ NEG	EN

Table 5: XR21B1420 Register Map

Address	Register Name	Bit 7 (15)	Bit 6 (14)	Bit 5 (13)	Bit 4 (12)	Bit 3 (11)	Bit 2 (10)	Bit 1 (9)	Bit 0 (8)
0x013	PIN_OPEN_DRAIN[15:8]	0	0	0	0	TX	0	GPIO9	GPIO8
	PIN_OPEN_DRAIN[7:0]	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x014	PIN_PULLUP_EN[15:8]	0	0	0	0	0	RX	GPIO9	GPIO8
	PIN_PULLUP_EN[7:0]	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x015	PIN_PULLDOWN_EN[15:8]	0	0	0	0	0	RX	GPIO9	GPIO8
	PIN_PULLDOWN_EN[7:0]	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x016	LOOPBACK	0	0	0	0	0	DTR_ DSR	RTS_ CTS	TX_ RX
0x017	IR_MODE	0	0	0	0	0	TX_- PULSE	RX_I NVERT	EN
0x018	OUTCLK[15:8]	DIV_HI							
	OUTCLK[7:0]	DIV_LO							
0x01F	REMOTE_WAKE	0	0	0	0	RX_EN	RI_EN	0	0
0x040	TX_FIFO_FLUSH	0	0	0	0	0	AUTO_- CLOSE	AUTO_ OPEN	RESET
0x041	TX_FIFO_COUNT[15:8]	0	0	0	0	0	0	COUNT[9:8]	
	TX_FIFO_COUNT[7:0]	COUNT[7:0]							
0x042	TX_WIDE_MODE	0	0	0	0	0	0	0	EN
0x043	RX_FIFO_FLUSH	0	0	0	0	0	AUTO_- CLOSE	AUTO_ OPEN	RESET
0x044	RX_FIFO_COUNT[15:8]	0	0	0	0	0	0	COUNT[9:8]	
	RX_FIFO_COUNT[7:0]	COUNT[7:0]							
0x045	RX_WIDE_MODE	0	0	0	0	0	0	0	EN
0x046	LOW_LATENCY	0	0	0	0	0	0	0	EN
0x047	RX_THRESHOLD[15:8]	0	0	0	0	0	0	COUNT[9:8]	
	RX_THRESHOLD[7:0]	COUNT[7:0]							
0x060	CUSTOM_DRIVER	0	0	0	0	0	0	0	ACTIVE
0x06A	SUSPEND_STATE[15:8]	0	0	DSR	DTR	RI	CD	0	0
	SUSPEND_STATE[7:0]	RXT	TXT	0	0	RS485	CTS	RTS	CLK
0x06B	SUSPEND_MODE[15:8]	USE_- SUS- PEND	0	DSR	DTR	RI	CD	0	0
	SUSPEND_MODE[7:0]	RXT	TXT	0	0	RS485	CTS	RTS	CLK
0x260	REVISION_ID[a]	VALUE							

Table 5: XR21B1420 Register Map

Address	Register Name	Bit 7 (15)	Bit 6 (14)	Bit 5 (13)	Bit 4 (12)	Bit 3 (11)	Bit 2 (10)	Bit 1 (9)	Bit 0 (8)
0x262	USB_STATUS[15:8][b]	0	0	STATE1	SEL1			CTRL1	
	USB_STATUS[7:0]	0	0	STATE0	SEL0			CTRL0	

a. The REVISION_ID register is accessed using XR_GET_REVISION, i.e. not XR_SET_REG or XR_GET_REG.
b. The USB_STATUS registers are accessed using XR_SET_USB_STAT and XR_GET_USB_STAT, i.e. not XR_SET_REG or XR_GET_REG.

© 2014 Exar Corporation

XR21B1420 Register Descriptions

UART_ENABLE (0x000) - Read/Write

The UART transmitter and receiver must be disabled before writing to any other UART registers except for the GPIO_SET, GPIO_CLEAR, TX_BREAK and ERROR_STATUS registers.

Bit	Default	Description
15:2	0	**Reserved** These bits are reserved and should be written as '0'.
1	0	**Enable UART RX** 0: UART RX disabled 1: UART RX enabled
0	0	**Enable UART TX** 0: UART TX disabled 1: UART TX enabled

FLOW_CONTROL (0x006) - Read/Write

This register selects the flow control mode. This register should only be written to when the UART is disabled. Writing to the FLOW_CONTROL register when the UART is enabled will result in undefined behavior.

Bit	Default	Description
15:4	0	**Reserved** These bits are reserved and should be written as '0'.
3	0	**UART Half-Duplex Mode** 0: Normal (full-duplex) mode. The UART can transmit and receive data at the same time. 1: UART Half-Duplex Mode. In half-duplex mode, any data on the RX pin is ignored when the UART is transmitting data.
2:0	000	**Mode** 000: Mode 0. No flow control, no address matching. 001: Mode 1. HW flow control enabled. Auto RTS/CTS or DTR/DSR must be selected by GPIO_MODE. 010: Mode 2. SW flow control enabled. 011: Mode 3. Multidrop mode - RX only after address match, TX independent. (Typically used with GPIO_-MODE 3). 100: Mode 4. Multidrop mode - RX/TX only after address match. (Typically used with GPIO_MODE 4). 101 to 111: Reserved

© 2014 Exar Corporation

XON_CHAR (0x007) - Read/Write

The XON_CHAR stores the 5 through 8 bit XON character that is used for Automatic Software Flow control. In 9 bit mode, only bits 7 through 0 are used, i.e. bit 8 is always a '0'. Alternately, this register holds the unicast address for multi-drop applications with address matching mode.

Bit	Default	Description
15:8	0	**Reserved** These bits are reserved and should be written as '0'.
7:0	0x11	**XON Character** In Automatic Software Flow control mode, the UART will suspend data transmission when the XON character has been received. For behavior in the address match mode, see "Multidrop mode with address matching" on page 14.

XOFF_CHAR (0x008) - Read/Write

The XOFF_CHAR stores the 5 through 8 bit XOFF character that is used for Automatic Software Flow control. In 9 bit mode, only bits 7 through 0 are used, i.e. bit 8 is always a '0'. Alternately, this register holds the multicast address for multi-drop applications with address matching mode.

Bit	Default	Description
15:8	0	**Reserved** These bits are reserved and should be written as '0'.
7:0	0x13	**XOFF Character** In Automatic Software Flow control mode, the UART will suspend data transmission when the XOFF character has been received. For behavior in the address match mode, see "Multidrop mode with address matching" on page 14.

ERROR_STATUS (0x009) - Read-Clear

This register reports any historical framing, parity and overrun errors as well as both current and historical break status, since the last time this register was read. As such, it does not indicate which character(s) the error(s) were associated with. For diagnostic purposes, WIDE_MODE may be enabled such that errors are directly associated with the current byte.

Bit	Default	Description
15:8	0	**Reserved** These bits are reserved and should be written as '0'.
7	0	**Break Status (Read-Only)** 0: Break condition is not present. 1: Break condition is currently being detected.
6	0	**Overrun Error** 0: No overrun error. 1: An overrun error has been detected (clears after read). An overrun error occurs when the RX FIFO is full and another byte of data is received.

© 2014 Exar Corporation

Bit	Default	Description
5	0	**Parity Error** 0: No parity error. 1: A parity error has been detected (clears after read).
4	0	**Framing Error** 0: No framing error. 1: A framing error has been detected (clears after read). A framing error occurs when a stop bit is not present when it is expected.
3	0	**Break Error** 0: No break condition. 1: A break condition has been detected (clears after read).
2:0	0	**Reserved** These bits are reserved and should be written as '0'.

TX_BREAK (0x00A) - Read/Write

This register controls UART TX break signaling.

Bit	Default	Description
15:0	0	**Value** For value TX_BREAK value of N: If N == 0xFFFF, the UART TX outputs a continuous break signal. If 0x0000 < N < 0xFFFF (a maximum of 64,534 ms), the UART TX outputs a break signal that lasts N ms, and the register serves as a counter, counting down to 0, decrementing by 1 every millisecond. If N == 0x0000, the UART TX stops sending the break signal. When the user writes to this register, any previous process is terminated, and the new command takes effect. If data is being shifted out of the TX pin, the data will be completely shifted out before the break condition is generated. NOTE: After this register is programmed from 0x0000 to a non-zero value, the UART TX may take up to, but no more than 1 ms, before sending out the break condition. In addition, after the break counter decrements to zero, the UART TX may take up to, but no more than 2 UART characters, based on the current UART configuration, before stopping the break. Thus, the actual break length may be slightly longer than the programmed value, by up to, but no more than (1ms + 2x UART-character-length).

XCVR_EN_DELAY (0x00B) - Read/Write

Bit	Default	Description
15:4	0	**Reserved** These bits are reserved and should be written as '0'.
3:0	0	**Turn-around delay** Turn-around delay controls the number of bit times (0-15) to wait before changing the direction of the RS-485 half-duplex from transmit to receive when auto RS-485 half-duplex control is enabled. This allows for propagation of characters to complete across lengthy mediums.

© 2014 Exar Corporation

GPIO_MODE (0x00C) - Read/Write

Bit	Default	Description
15:10	0	**Reserved** These bits are reserved and should be written as '0'.
9	1	**Receive Toggle** 0: GPIO9 is used for general purpose I/O 1: GPIO9 is used to receive toggle output (default).
8	1	**Transmit Toggle** 0: GPIO8 is used for general purpose I/O. 1: GPIO8 is used to transmit toggle output (default).
7	0	**Clock Enable** 0: GPIO6 is used for general purpose I/O 1: GPIO6 is used to output a clock. See "OUTCLK (0x018) - Read/Write" on page 31.
6:5	0	**Auto RS-485 Half-Duplex Select** 00: GPIO. GPIO7/RS485 is used for general purpose I/O 01: RS485_EN_ACT. GPIO7/RS485 is used for auto RS-485 half-duplex enable. Asserted whenever the UART is transmitting 10: RS485_EN_FLOW. GPIO7/RS485 is used for auto RS-485 half-duplex enable. Asserted for the duration of the address match 11: RESERVED. Reserved value, do not use
4	0	**Auto RS-485 Half-Duplex Pin** 0: GPIO5/RTS#/RS485 function is selected by GPIO_MODE[2:0]. GPIO7/RS485 function is GPIO. 1: GPIO7/RS485 function is selected by GPIO_MODE[6:5]. GPIO5/RTS#/RS485 function must be any function other than that selected for GPIO7/RS485.
3	0	**Auto RS-485 Half-Duplex Polarity** 0: Active low auto. RS-485 half-duplex enable 1: Active high auto. RS-485 half-duplex enable
2:0	000	**GPIO Mode Select** 000: GPIO. RTS/CTS and DTR/DSR are used for general purpose I/O. 001: RTS_CTS. GPIO4 and GPIO5 used for Auto RTS/CTS HW Flow Control 010: DTR_DSR. GPIO2 and GPIO3 used for Auto DTR/DSR HW Flow Control 011: RS485_EN_ACT. GPIO5/RTS#/RS485 pin used for auto RS-485 half-duplex enable during Transmit 100: RS485_EN_FLOW. GPIO5/RTS#/RS485 pin used for auto RS-485 half-duplex enable after address match. 101 to 111: Reserved. Reserved value, do not use.

© 2014 Exar Corporation

GPIO_DIRECTION (0x00D) - Read/Write

This register controls the direction of pins that are configured as GPIO. Pins that are configured for alternate functions via the GPIO_MODE register are not controlled by this register.

Bit	Default	Description
15:10	0	**Reserved** These bits are reserved and should be '0'.
9:0	0	**GPIO Direction of GPIO[9:0]** 0: GPIOx is an input. 1: GPIOx is an output.

GPIO_SET (0x00E) - Write-Only

This register controls pins configured as GPIO outputs. Pins configured for alternate functions via the GPIO_MODE register are not controlled by this register. Writing a '1' to a bit position in this register sets the corresponding GPIO output high. Writing a '0' to a bit has no effect. For GPIO pins configured as inputs via the GPIO_DIRECTION register, this register has no effect.

Bit	Default	Description
15:10	0	**Reserved** These bits are reserved and should be '0'.
9:0	0	**GPIO Set of GPIO[9:0]** 0: No effect on GPIOx pin. 1: GPIOx output is set to a '1'.

GPIO_CLEAR (0x00F) - Write-Only

This register controls pins configured as GPIO outputs. Pins configured for alternate functions via the GPIO_MODE register are not controlled by this register. Writing a '1' to a bit position in this register clears the corresponding GPIO output low. Writing a '0' to a bit has no effect. For GPIO pins configured as inputs via the GPIO_DIRECTION register, this register has no effect.

Bit	Default	Description
15:10	0	**Reserved** These bits are reserved and should be '0'.
9:0	0	**GPIO Set of GPIO[9:0]** 0: No effect on GPIOx pin. 1: GPIOx output is cleared to a '0'.

© 2014 Exar Corporation