



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



### General Description

The XR21B1422 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 XR21B1422 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 for each port 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 XR21B1422 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 XR21B1422 operates from a single 5V or 3.3V power supply. When powered with 5V input, a regulated 3.3V output is supplied.

### FEATURES

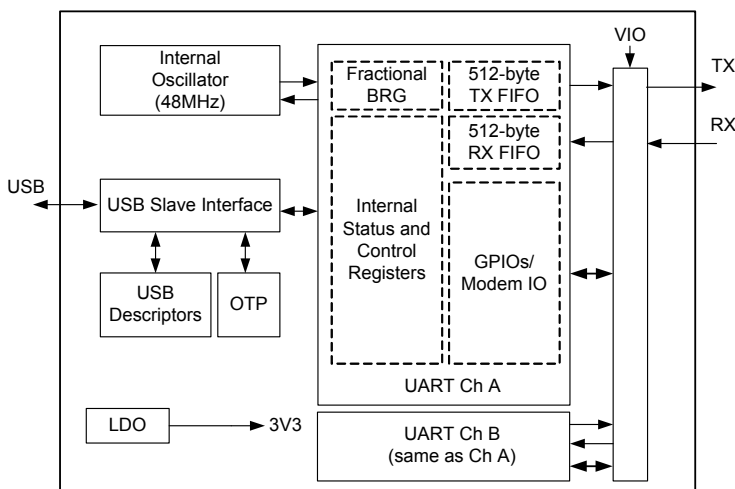
- ±15kV ESD on USB D+/USB D-
- USB 2.0 Compliant, Full-Speed (12Mbps)
- Unique pre-programmed USB serial number
- Internally generated 48MHz core clock
- Enhanced UART features
  - Baud rates up to 12 Mbps
  - Fractional Baud Rate Generator
  - 512-byte TX and 512-byte RX FIFOs
  - Auto Hardware / Software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

### APPLICATIONS

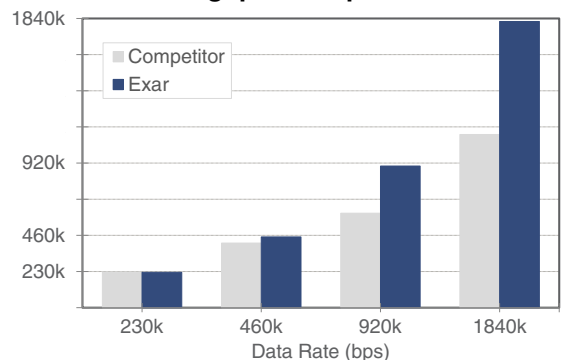
- Building Automation
- Security Systems
- Factory and Process Control
- ATM Terminals
- USB to Serial Controllers

Ordering Information – [page 60](#)

### Block Diagram



Throughput Comparison



## 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, VIO).....	+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

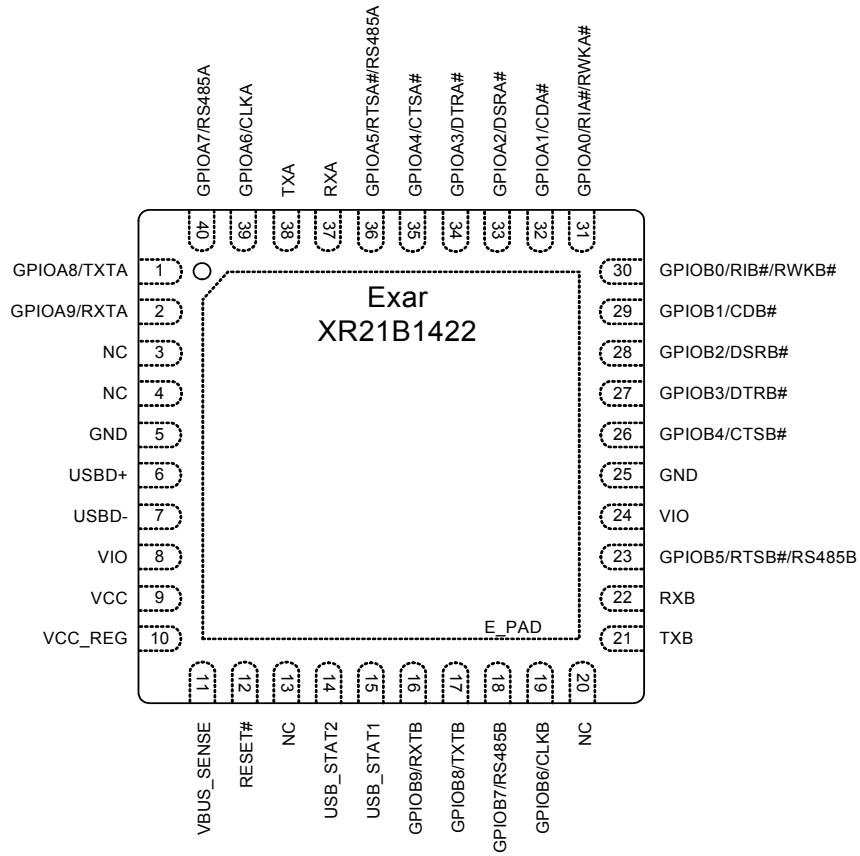
## Electrical Characteristics

Unless otherwise noted:  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ , VCC\_REG = +4.4V to +5.25V or +3.0V to +3.6V, VIO = +1.8V to +3.6V.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
Power						
I <sub>CC</sub>	Power Supply Current	VCC_REG = +4.4V to +5.25V		15	23	mA
I <sub>SUSP</sub>	Lowpower Mode Current			0.85	1.25	mA
V <sub>OUT</sub>	Regulated Output Voltage (VCC pin)	VCC_REG = +4.4V to +5.25V. Maximum output current = 200 mA including the supply current of the XR21B1422.	3	3.3	3.6	V
UART, USB_STAT and GPIO Pins						
V <sub>IL</sub>	Input Low Voltage		-0.3		0.25* VIO	V
V <sub>IH</sub>	Input High Voltage		0.70* VIO		5.5	V
V <sub>OL</sub>	Output Low Voltage	IOL = 1mA, VIO = +1.6V			0.3	V
		IOL = 4mA, VIO = +3.6V			0.5	V
V <sub>OH</sub>	Output High Voltage	IOH = -400uA, VIO = +1.6V	1.3		VIO	V
		IOH = -1.5mA, VIO = +3.6V	2.8		VIO	V
I <sub>IL</sub>	Input Low Leakage Current	VIO = +3V to +3.6V, VCC_REG = +4.4V to +5.25V, V <sub>INPUT</sub> = 0V			±10	µA
I <sub>IH</sub>	Input High Leakage Current	VIO = +3V to +3.6V, VCC_REG = +4.4V to +5.25V, V <sub>INPUT</sub> = +3.3V			±10	µA
		VIO = +3V to +3.6V, VCC_REG = +4.4V to +5.25V, V <sub>INPUT</sub> = +5.5V			±120	µA
C <sub>IN</sub>	Input Pin Capacitance				5	pF

Symbol	Parameter	Conditions	Min	Typ	Max	Units
USB I/O Pins						
V <sub>IL</sub>	Input Low Voltage		-0.3		0.8	V
V <sub>IH</sub>	Input High Voltage		2.0		5.5	V
V <sub>OL</sub>	Output Low Voltage	External 15k $\Omega$ to GND on USB <sub>D+</sub> and USB <sub>D-</sub> pins	0		0.3	V
V <sub>OH</sub>	Output High Voltage	External 15k $\Omega$ to GND on USB <sub>D+</sub> and USB <sub>D-</sub> pins	2.8		3.6	V
V <sub>DrvZ</sub>	Driver Output Impedance		28		44	$\Omega$

Pin Configuration



## Pin Assignments

Pin No.	Pin Name	Type	Description
1	GPIOA8/TXTA	I/O	General purpose I/O, or UART transmit data indicator. Defaults to GPIO input with internal pull-up resistor. See “ <a href="#">TXT and RXT Pins</a> ” on page 18. When configured as transmit indicator, this pin will toggle at ~10Hz intervals while the UART is transmitting data.
2	GPIOA9/RXTA	I/O	General purpose I/O, or UART receive data indicator. Defaults to GPIO input with internal pull-up resistor. See “ <a href="#">TXT and RXT Pins</a> ” on page 18. When configured as receive indicator, this pin will toggle at ~10Hz intervals while the UART is receiving data.
3	NC	-	No Connect.
4	NC	-	No Connect.
5	GND	PWR	Power supply common, ground.
6	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.
7	USBD-	I/O	USB port differential data negative. The ESD protection on this pin is ±15kV HBM.
8	VIO	PWR	I/O voltage input to the UART/GPIO pins. Must be between 1.8 and 3.6V. May be connected to VCC (pin 12) if VCC_REG input voltage is 5V. See “ <a href="#">USB Power Modes</a> ” on page 11.
9	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 “ <a href="#">USB Power Modes</a> ” on page 11.
10	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 <a href="#">Figure 1</a> . 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 <a href="#">Figure 2</a> and <a href="#">Figure 3</a> .
11	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.
12	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.
13	NC	-	No Connect.
14	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 XR21B1422 is placed into a suspended state. This default may be changed via the PIN_CFG_USB_STAT2 register.
15	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 XR21B1422 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 XR21B1422 is placed into a suspended state.

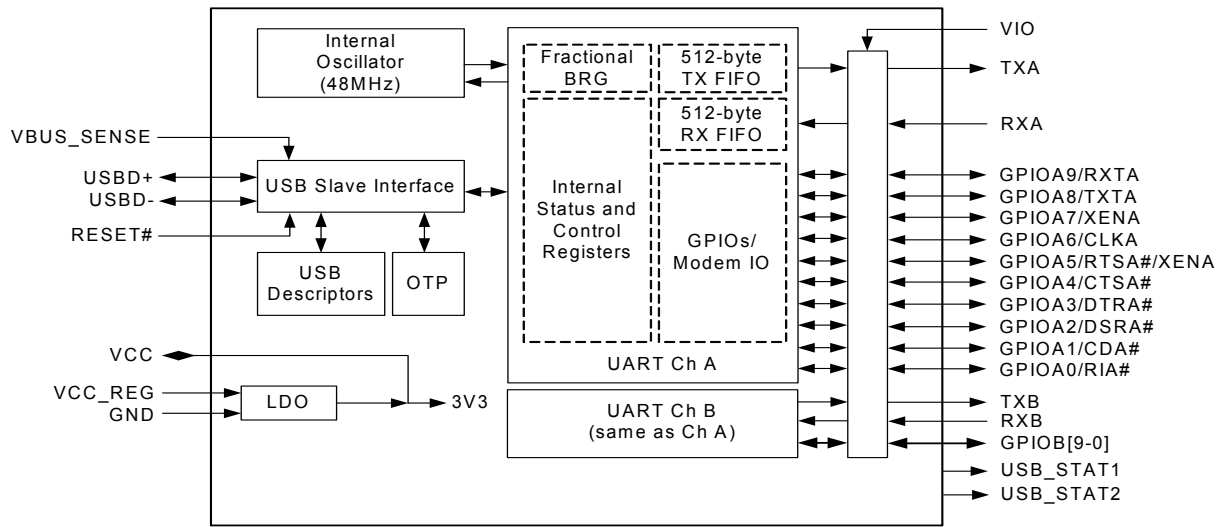
Pin No.	Pin Name	Type	Description
16	GPIOB9/RXTB	I/O	General purpose I/O, or UART receive data indicator. Defaults to GPIO input with internal pull-up resistor. See <a href="#">“TXT and RXT Pins” on page 18</a> . When configured as receive indicator, this pin will toggle at ~10Hz intervals while the UART is receiving data.
17	GPIOB8/TXTB	I/O	General purpose I/O, or UART transmit data indicator. Defaults to GPIO input with internal pull-up resistor. See <a href="#">“TXT and RXT Pins” on page 18</a> . When configured as transmit indicator, this pin will toggle at ~10Hz intervals while the UART is transmitting data.
18	GPIOB7/RS485B	I/O	General purpose I/O, or auto RS-485 half-duplex control. Defaults to GPIO input with internal pull-up resistor.
19	GPIOB6/CLKB	I/O	General purpose I/O, or clock or pulse output. Defaults to GPIO input with internal pull-up resistor. See <a href="#">“Programmable Output Clock” on page 15</a> .
20	NC	-	No Connect.
21	TXB	O	UART Transmit Data.
22	RXB	I	UART Receive Data.
23	GPIOB5/RTSB#/RS485B	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 <a href="#">“Automatic RTS/CTS Hardware Flow Control” on page 15</a> or <a href="#">“Multidrop mode with address matching” on page 17</a> .
24	VIO	PWR	I/O voltage input to the UART/GPIO pins. Must be between 1.8 and 3.6V. May be connected to VCC (pin 12) if VCC_REG input voltage is 5V. See <a href="#">“USB Power Modes” on page 11</a> .
25	GND	PWR	Power supply common, ground.
26	GPIOB4/CTSB#	I/O	General purpose I/O, or UART Clear-to-Send input (active low). Defaults to GPIO input with internal pull-up resistor. See <a href="#">“Automatic RTS/CTS Hardware Flow Control” on page 15</a> .
27	GPIOB3/DTRB#	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 <a href="#">“Automatic DTR/DSR Hardware Flow Control” on page 16</a> .
28	GPIOB2/DSRB#	I/O	General purpose I/O, or UART Data-Set-Ready input (active low). Defaults to GPIO input with internal pull-up resistor. See <a href="#">“Automatic DTR/DSR Hardware Flow Control” on page 16</a> .
29	GPIOB1/CDB#	I/O	General purpose I/O, or UART Carrier-Detect input (active low). Defaults to GPIO input with internal pull-up resistor.
30	GPIOB0/RIB#/RWKB#	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 <a href="#">“Remote Wakeup” on page 9</a> .
31	GPIOA0/RIA#/RWKA#	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 <a href="#">“Remote Wakeup” on page 9</a> .
32	GPIOA1/CDA#	I/O	General purpose I/O, or UART Carrier-Detect input (active low). Defaults to GPIO input with internal pull-up resistor.
33	GPIOA2/DSRA#	I/O	General purpose I/O, or UART Data-Set-Ready input (active low). Defaults to GPIO input with internal pull-up resistor. See <a href="#">“Automatic DTR/DSR Hardware Flow Control” on page 16</a> .
34	GPIOA3/DTRA#	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 <a href="#">“Automatic DTR/DSR Hardware Flow Control” on page 16</a> .
35	GPIOA4/CTSA#	I/O	General purpose I/O, or UART Clear-to-Send input (active low). Defaults to GPIO input with internal pull-up resistor. See <a href="#">“Automatic RTS/CTS Hardware Flow Control” on page 15</a> .
36	GPIOA5/RTSA#/RS485A	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 <a href="#">“Automatic RTS/CTS Hardware Flow Control” on page 15</a> or <a href="#">“Multidrop mode with address matching” on page 17</a> .
37	RXA	I	UART Receive Data.

Pin No.	Pin Name	Type	Description
38	TXA	O	UART Transmit Data.
39	GPIOA6/CLKA	I/O	General purpose I/O, or clock or pulse output. Defaults to GPIO input with internal pull-up resistor. See <a href="#">“Programmable Output Clock”</a> on page 15.
40	GPIOA7/RS485A	I/O	General purpose I/O, or auto RS-485 half-duplex control. Defaults to GPIO input with internal pull-up resistor. See <a href="#">“Multidrop mode with address matching”</a> on page 17.

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



Functional Block Diagram



## Functional Description

### USB Interface

The USB interface of the XR21B1422 is compliant with the USB 2.0 Full-Speed Specifications.

The XR21B1422 uses the following set of parameters:

- 1 Control Endpoint
  - Endpoint 0 as outlined in the USB specifications
- 1 Configuration is supported
- 1 Interface for each UART channel
  - 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 XR21B1422. Customers may obtain their own Vendor ID from USB.org. The default USB Product ID for the XR21B1422 is 0x1422. 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 XR21B1422 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 XR21B1422) that draws power from the host VBUS.

### Remote Wakeup

If the XR21B1422 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 XR21B1422 device reports in its USB device attributes that it supports remote wakeup. The RI#/RWK# pins of each UART channel are enabled for remote wakeup signaling if their default configuration as an input pin has not been changed. The RI#/RWK# pins from each UART channel are logically ANDed, such that a logic '0' on any of the two pins will prevent the remote wakeup signaling. Additionally, the RX pins of each UART channel may also be enabled via OTP to support remote wakeup. Again all RX pins that are enabled to support remote wakeup signaling are also logically ANDed. 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 XR21B1422 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.

**Table 1: USB String Descriptor Defaults**

Descriptor	Value
Exar USB Vendor ID	0x04E2
Exar USB Product ID	0x1422
Manufacturer String	Exar Corp.
Product String	Exar USB UART

## Device Driver

The XR21B1422 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 XR21B1422, 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 XR21B1422 internal device registers, the device is initialized to certain hardware defaults. By default the XR21B1422 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 14](#). This threshold may be modified in the OTP CDC\_ACM\_BAUD\_THRESH locations.

**Table 2: XR21B1422 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 XR21B1422 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 XR21B1422 automatically changing to the settings in the [Table 2](#).

## USB Power Modes

The XR21B1422 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 XR21B1422 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 XR21B1422 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 XR21B1422 device. The VCC pin will supply a 3.3V output. The VIO pins may be externally connected to VCC or to an alternate voltage source.

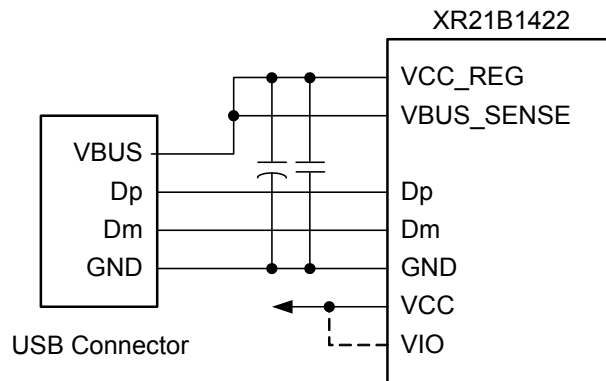


Figure 1: Bus-Powered Mode

### Self-Powered 5V

In self-powered 5V mode, a local source provides 5V to the XR21B1422 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. VIO pins may be externally connected to VCC or to an alternate voltage source.

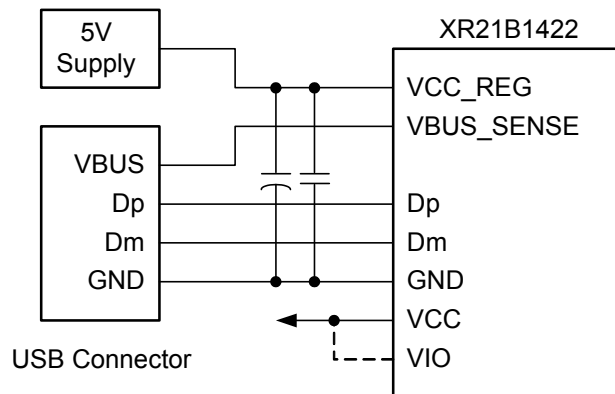


Figure 2: Self-Powered 5V Mode

## 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 XR21B1422 device. The USB attributes should be changed in the OTP to correctly report self-powered mode. VIO pins may be externally connected to VCC or to an alternate voltage source.

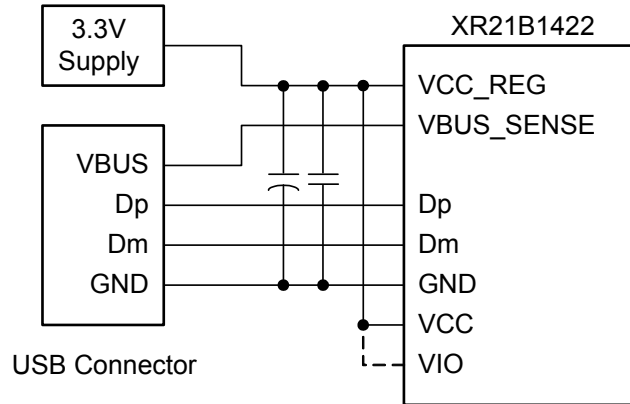


Figure 3: Self-Powered 3.3V Mode

## Reset

The XR21B1422 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 XR21B1422 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 XR21B1422, these defaults will be changed. See [“Device Driver”](#) on page 10.

## 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](#).

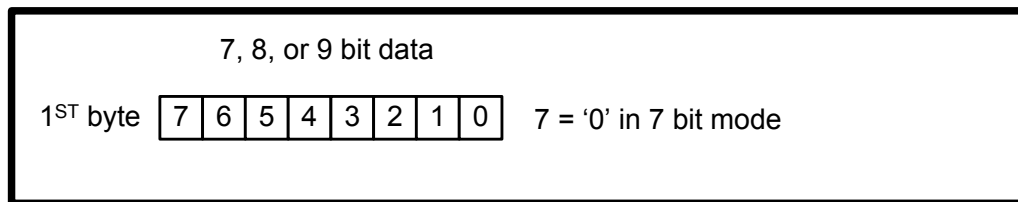


Figure 4: Receive Data Format

### 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](#).

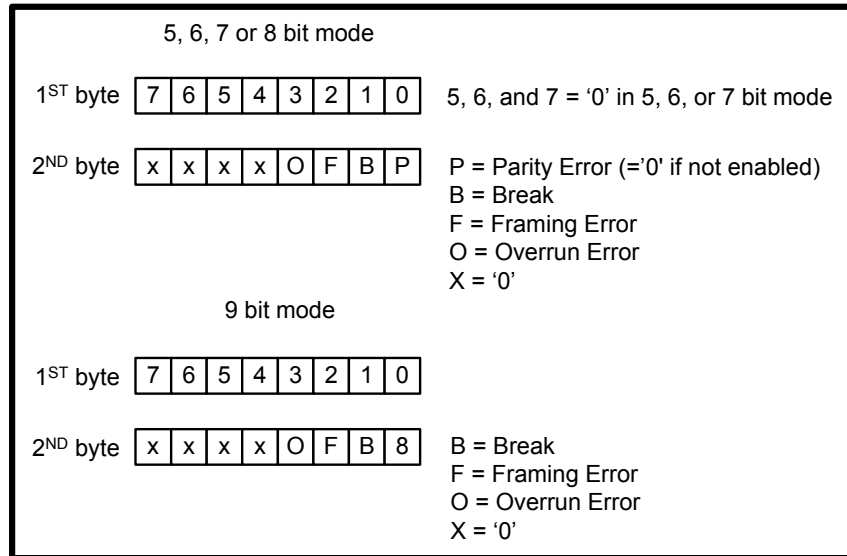


Figure 5: Wide Mode Receive Data Format

### 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 XR21B1422 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 XR21B1422 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 XR21B1422 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 XR21B1422 has a low latency register bit that will enable the XR21B1422 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.

### 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:

$$\text{FREQ} = 24 \text{ MHz} / (\text{DIV\_HI} + \text{DIV\_LO}). \text{ The duty cycle is determined by the ratio of DIV\_HI to DIV\_LO.}$$

### Flow Control

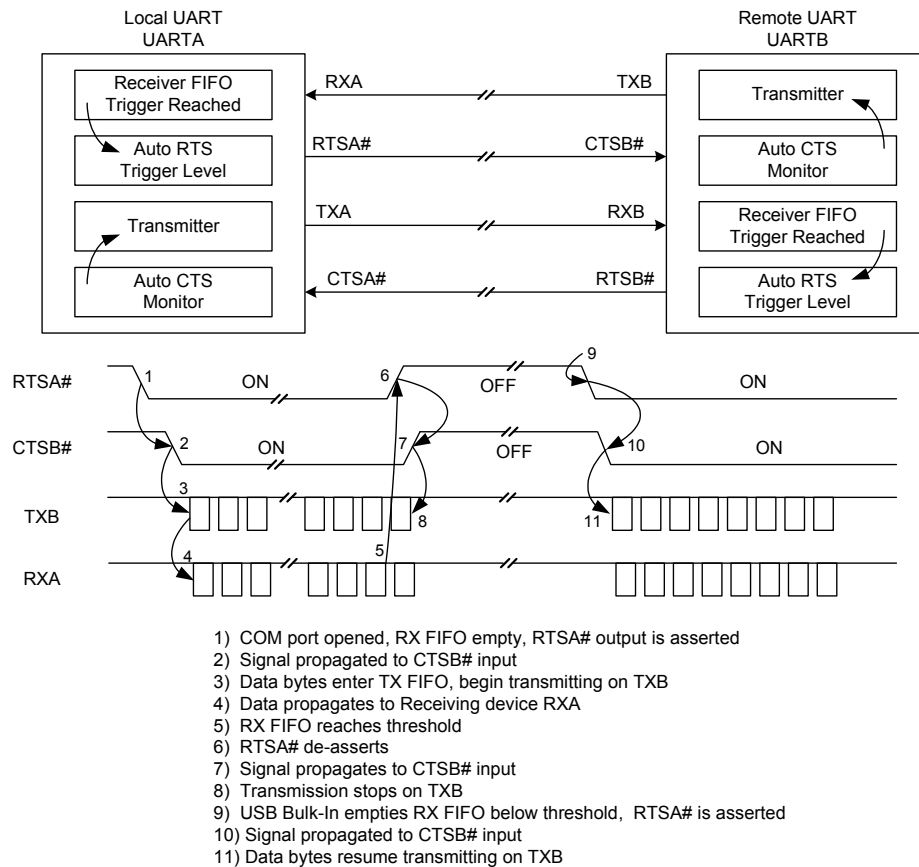
The XR21B1422 is able to perform both hardware and software flow control. Both hardware and software flow control modes are configured via the GPIO\_MODE and FLOW\_CONTROL registers. In both modes, flow control is asserted when the bytes in the RX FIFO reach the watermark set in the RX\_THRESHOLD register.

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 B1422 CTS input will cause the device to suspend data transmission.





**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 XR21B1422 compares the receive data characters with the programmed XON or XOFF characters. If the received character matches the programmed XOFF character, the XR21B1422 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 XR21B1422 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.

## IR Mode

The XR21B1422 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 XR21B1422 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 XR21B1422 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 XR21B1422 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 XR21B1422 device.

## Suspend Mode Settings

The USE\_SUSPEND bit controls the GPIO pins when the XR21B1422 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.

## USB Control Commands

The following table shows all of the USB Control Commands that are supported by the XR21B1422. 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_FEATURE	0x00	0x1	0x1	0x0	0x0	0x0	0x0	0x0	Device remote wake-up
EP_CLEAR_FEATURE	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_CONFIGURATION	0x80	0x8	0x0	0x0	0x0	0x0	0x1	0x0	
SET_CONFIGURATION	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	Chan #	0x0	0x7	0x0	Set the UART baud rate, parity, stop bits, etc. Channel #0, 2 for channel A, B respectively.
CDC_ACM_IF GET_LINE_CODING	0xA1	0x21	0x0	0x0	Chan #	0x0	0x7	0x0	Get the UART baud rate, parity, stop bits, etc. Channel #0, 2 for channel A, B respectively.
CDC_ACM_IF SET_CONTROL_LINE_STATE	0x21	0x22	0x0	0x0	Chan #	0x0	0x7	0x0	Set/Clear DTR in CDC-ACM mode. Channel #0, 2 for channel A, B respectively.
CDC_ACM_IF SEND_BREAK	0x21	0x23	val LSB	val MSB	Chan #	0x0	0x0	0x0	Send a break for the specified duration. Channel #0, 2 for channel A, B respectively.
XR_GET_CHIP_ID	0xC0	0xFF	0x0	0x0	0x0	0x0	0x6	0x0	Get Exar VID (2 bytes), PID (2 bytes) and bcdDevice (2 bytes)

Table 4: Supported USB Control Commands

Name	Request Type	Request	Value		Index		Length		Description
			LSB	MSB	LSB	MSB	LSB	MSB	
XR_SET_REG See Table 5	0x41	0x0	write- data LSB	write- data MSB	write addr	Chan #	0x0	0x0	Vendor specific register access. Channel #0, 2 for channel A, B respectively.
XR_GET_REG See Table 5	0xC1	0x0	0x0	0x0	read addr	Chan #	0x2	0x0	Vendor specific register access. Channel #0, 2 for channel A, B respectively.
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.

## Register Set Description

The internal register set of the XR21B1422 controls the UART channel 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: XR21B1422 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: XR21B1422 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_ INVERT	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 <sup>a</sup>	VALUE							

Table 5: XR21B1422 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] <sup>b</sup>	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.



## XR21B1422 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	<b>Reserved</b> These bits are reserved and should be written as '0'.
1	0	<b>Enable UART RX</b> 0: UART RX disabled 1: UART RX enabled
0	0	<b>Enable UART TX</b> 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	<b>Reserved</b> These bits are reserved and should be written as '0'.
3	0	<b>UART Half-Duplex Mode</b> 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	<b>Mode</b> 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

### 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	<b>Reserved</b> These bits are reserved and should be written as '0'.

Bit	Default	Description
7:0	0x11	<b>XON Character</b> 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 17.

### 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	<b>Reserved</b> These bits are reserved and should be written as '0'.
7:0	0x13	<b>XOFF Character</b> 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 17.

### 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	<b>Reserved</b> These bits are reserved and should be written as '0'.
7	0	<b>Break Status (Read-Only)</b> 0: Break condition is not present. 1: Break condition is currently being detected.
6	0	<b>Overrun Error</b> 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.
5	0	<b>Parity Error</b> 0: No parity error. 1: A parity error has been detected (clears after read).
4	0	<b>Framing Error</b> 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	<b>Break Error</b> 0: No break condition. 1: A break condition has been detected (clears after read).
2:0	0	<b>Reserved</b> These bits are reserved and should be written as '0'.