GENERAL DESCRIPTION

The XR21V1412 (V1412) is an enhanced 2-channel Universal Asynchronous Receiver and Transmitter (UART) with a USB interface. The USB interface is fully compliant to Full Speed USB 2.0 specification that supports 12 Mbps USB data transfer rate. The USB interface also supports USB suspend, resume and remote wakeup operations.

The V1412 operates from an internal 48MHz clock therefore no external crystal/oscillator is required like previous generation UARTs. With the fractional baud rate generator, any baud rate can accurately be generated using the internal 48MHz clock.

The large 128-byte FIFO and 384-byte RX FIFO of the V1412 helps to optimize the overall data throughput for various applications. The Automatic Transceiver Direction control feature simplifies both the hardware and software for half-duplex RS-485 applications. If required, the multidrop (9-bit) mode with automatic half-duplex transceiver control feature further simplifies typical multidrop RS-485 applications.

The V1412 operates from a single 2.97 to 3.63 volt power supply and has 5V tolerant inputs. The V1412 is available in a 32-pin QFN package.

Software drivers for Windows 2000, XP, Vista, and CE, as well as Linux and Mac are supported for the XR21V1412.

APPLICATIONS

- Portable Appliances
- External Converters (dongles)
- Battery-Operated Devices
- Cellular Data Devices
- Factory Automation and Process Controls
- Industrial applications

FEATURES

- USB 2.0 Compliant Interface
  - Supports 12 Mbps USB full-speed data rate
  - Supports USB suspend, resume and remote wakeup operations
- Enhanced Features of each UART
  - Data rates up to 12 Mbps
  - Fractional Baud Rate Generator
  - 128 byte TX FIFO
  - 384 byte RX FIFO
  - 7, 8 or 9 data bits, 1 or 2 stop bits
  - Automatic Hardware (RTS/CTS or DTR/DSR) Flow Control
  - Automatic Software (Xon/Xoff) Flow Control
  - Multidrop mode w/ Auto Half-Duplex Transceiver Control
  - Multidrop mode w/ Auto TX Enable
  - Half-Duplex mode
  - Sleep Mode with Remote Wake-up
  - Selectable GPIO or Modem I/O
- Internal 48 MHz clock
- Single 2.97-3.63V power supply
- 5V tolerant inputs
- 32-pin QFN package
- Virtual COM Port drivers
  - Windows 2000, XP and Vista
  - Windows CE 4.2, 5.0, 6.0
  - Linux
  - Mac

FIGURE 1. XR21V1412 BLOCK DIAGRAM

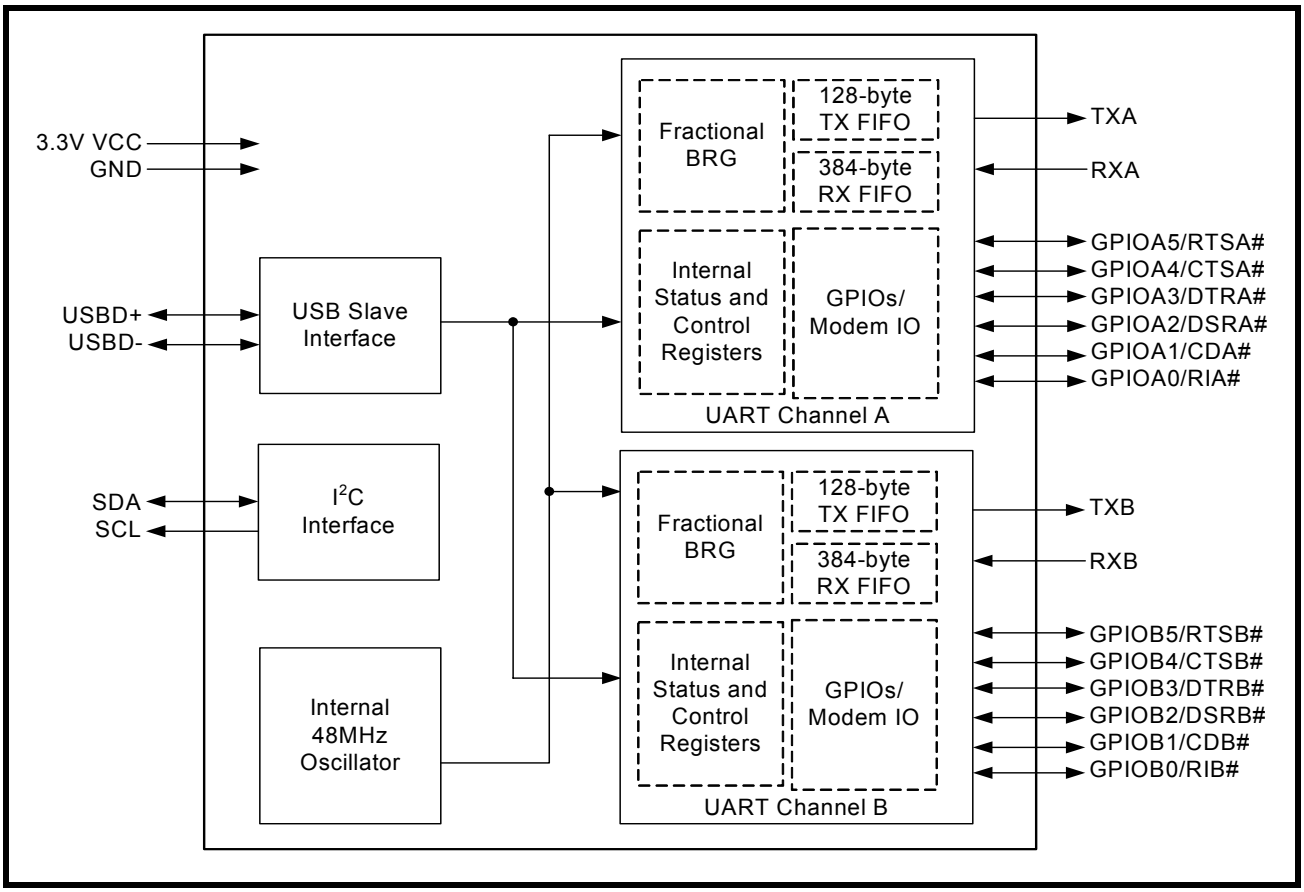
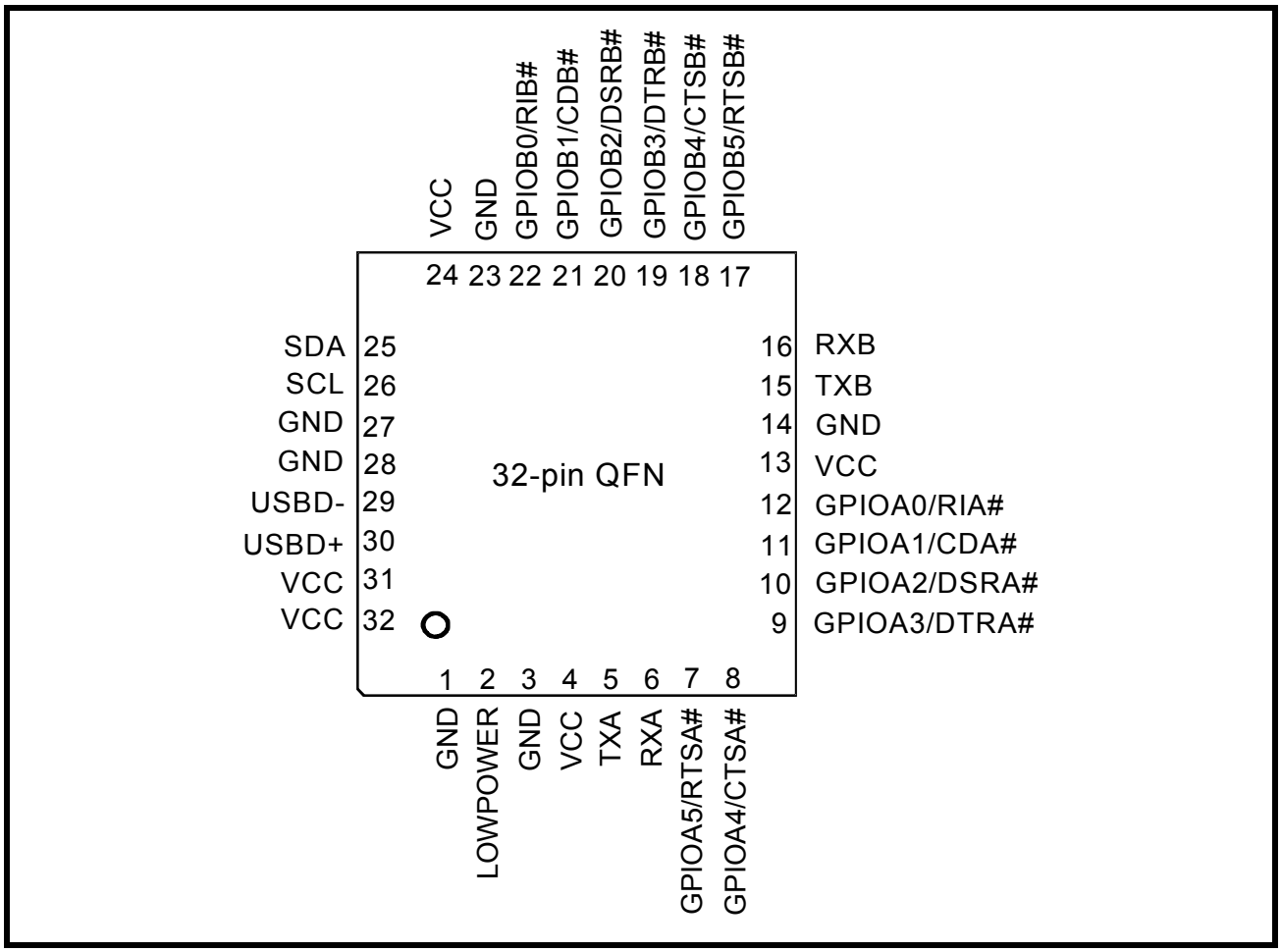


FIGURE 2. PIN OUT ASSIGNMENT



ORDERING INFORMATION

PART NUMBER	PACKAGE	OPERATING TEMPERATURE RANGE	DEVICE STATUS
XR21V1412IL32	32-pin QFN	-40°C to +85°C	Active

PIN DESCRIPTIONS

Pin Description

NAME	32-QFN PIN #	TYPE	DESCRIPTION
UART Channel A Signals			
RXA	6	I	UART Channel A Receive Data or IR Receive Data. This pin has an internal pull-up resistor. Internal pull-up resistor is not disabled during suspend mode.
TXA	5	O	UART Channel A Transmit Data or IR Transmit Data.
GPIOA0/RIA#	12	I/O	General purpose I/O or UART Ring-Indicator input (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOA1/CDA#	11	I/O	General purpose I/O or UART Carrier-Detect input (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOA2/DSRA#	10	I/O	General purpose I/O or UART Data-Set-Ready input (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOA3/DTRA#	9	I/O	General purpose I/O or UART Data-Terminal-Ready output (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOA4/CTSA#	8	I/O	General purpose I/O or UART Clear-to-Send input (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOA5/RTSA#	7	I/O	General purpose I/O or UART Request-to-Send output (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
UART Channel B Signals			
RXB	16	I	UART Channel B Receive Data or IR Receive Data. This pin has an internal pull-up resistor. Internal pull-up resistor is not disabled during suspend mode.
TXB	15	O	UART Channel B Transmit Data or IR Transmit Data.
GPIOB0/RIB#	22	I/O	General purpose I/O or UART Ring-Indicator input (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOB1/CDB#	21	I/O	General purpose I/O or UART Carrier-Detect input (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOB2/DSRB#	20	I/O	General purpose I/O or UART Data-Set-Ready input (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.

Pin Description

NAME	32-QFN PIN #	TYPE	DESCRIPTION
GPIOB3/DTRB#	19	I/O	General purpose I/O or UART Data-Terminal-Ready output (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOB4/CTSB#	18	I/O	General purpose I/O or UART Clear-to-Send input (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
GPIOB5/RTSB#	17	I/O	General purpose I/O or UART Request-to-Send output (active low). This pin has an internal pull-up resistor. Internal pull-up resistor is disabled during suspend mode. If using this GPIO as an input, a pull-up resistor is required to minimize the power consumption in the suspend mode.
USB Interface Signals			
USBD+	30	I/O	USB port differential data plus. This pin has a 1.5 K Ohm internal pull-up resistor.
USBD-	29	I/O	USB port differential data minus.
I2C Interface Signals			
SDA	25	OD	I^2C-controller data input/output (open-drain). 1K pull-up resistor is required. An optional external I^2C EEPROM can be used to store default configurations upon power-up including the USB Vendor ID and Device ID. If an EEPROM is not used, this pin can be used with the SCL pin to select the Remote Wake-up and Power modes. An external pull-up or pull-down resistor is required. See **Table 1**.
SCL	26	I	I^2C-controller serial input clock. 1K pull-up resistor is required. An optional external I^2C EEPROM can be used to store default configurations upon power-up including the USB Vendor ID and Device ID. If an EEPROM is not used, this pin can be used with the SDA pin to select the Remote Wake-up and Power modes. An external pull-up or pull-down resistor is required. See **Table 1**.
Ancillary Signals			
LOWPOWER	2	O	Low power status output. This pin is HIGH when the XR21V1412 is in the suspend mode. This pin is LOW when the XR21V1412 is not in the suspend mode. An external pull-up or pull-down resistor is required on this pin. This pin is sampled upon power-on to configure the polarity of the LOWPOWER output during suspend mode. An external pull-up resistor will cause the LOWPOWER pin to be HIGH during suspend mode. An external pull-down resistor will cause the LOWPOWER pin to be LOW during suspend mode
VCC	4, 13, 24, 31, 32	Pwr	+3.3V power supply. All inputs are 5V tolerant.
GND	1, 3, 14, 23, 27, 28	Pwr	Power supply common, ground.

NOTE: **Pin type: I=Input, O=Output, I/O= Input/output, OD=Output Open Drain.**

1.0 FUNCTIONAL DESCRIPTIONS

1.1 USB interface

The USB interface of the V1412 is compliant with the USB 2.0 Full-Speed Specifications. The USB configuration model presented by the V1412 to the device driver is compatible to the Abstract Control Model of the USB Communication Device Class (CDC-ACM). The V1412 uses the following set of parameters:

- 1 Control Endpoint
  - Endpoint 0 as outlined in the USB specifications
- 1 Configuration is supported
- 2 interfaces per UART channel
  - Each UART channel has a single interrupt endpoint
  - Each UART channel have bulk-in and bulk-out endpoints

1.1.1 USB Vendor ID

Exar's USB Vendor ID is 0x04E2. This is the default Vendor ID that is used for the V1412 unless a valid EEPROM is present on the I2C interface signals. If a valid EEPROM is present, the Vendor ID from the EEPROM will be used.

1.1.2 USB Product ID

The default USB Product ID for the V1412 is 0x1412. If a valid EEPROM is present, the Product ID from the EEPROM will be used.

1.2 I2C Interface

The I2C interface provides connectivity to an external I2C memory device (i.e. EEPROM) that can be read by the V1412 for configuration.

The SDA and SCL are used to specify whether Remote Wakeup and/or Bus Powered configurations are to be supported. These pins are sampled at power-up. The following table describes how Remote Wakeup and Bus Powered support.

TABLE 1: REMOTE WAKEUP AND POWER MODES

SDA	SCL	REMOTE WAKE-UP SUPPORT	POWER MODE	COMMENTS
1	1	No	Self-Powered	Default, if no EEPROM is present
1	0	No	Bus-Powered	
0	1	Yes	Self-Powered	
0	0	Yes	Bus-Powered	

1.2.1 EEPROM Contents

The I2C address should be 0xA0. An EEPROM can be used to override default Vendor IDs and Device IDs, as well as other attributes and maximum power consumption. The EEPROM must contain 8 bytes of data as specified in Table 2

TABLE 2: EEPROM CONTENTS

EEPROM ADDRESS	CONTENTS
0	Vendor ID (LSB)
1	Vendor ID (MSB)
2	Product ID (LSB)
3	Product ID (MSB)
4	Device Attributes
5	Device Maximum Power
6	Reserved
7	Signature of 0x58 ('X'). If the signature is not correct, the contents of the EEPROM are ignored.

These values are uploaded from the EEPROM to the corresponding USB Standard Device Descriptor or Standard Configuration Descriptor. For details of the USB Descriptors, refer to the USB 2.0 specifications.

1.2.1.1 Vendor ID

The Vendor ID value replaces the idVendor field in the USB Standard Device Descriptor.

1.2.1.2 Product ID

The Product ID value replaces the idProduct field in the USB Standard Device Descriptor.

1.2.1.3 Device Attributes

The Device Attributes value replaces the bmAttributes field in the USB Standard Configuration Descriptor.

1.2.1.4 Device Maximum Power

The Device Maximum Power value replaces the bMaxPower field in the USB Standard Configuration Descriptor.

1.3 UART Manager

The UART Manager enables/disables each UART including the TX and RX FIFOs for each UART. The UART Manager is located in a separate register block from the 2 UART channels.

1.4 UART

There are 2 enhanced UART channels in the V1412. Each UART channel is independent, therefore, they will need to be initialized and configured independently. Each UART can be configured via USB control transfers from the USB host.

1.4.1 Transmitter

The transmitter consists of a 128-byte TX FIFO and a Transmit Shift Register (TSR). Once a bulk-out packet has been received and the CRC has been validated, the data bytes in that packet are written into the TX FIFO of the specified UART channel. 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 data rate defined by the CLOCK_DIVISOR and TX_CLOCK_MASK registers. 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 can be configured for 7 or 8 data bits with parity or 9 data bits with no parity.

1.4.1.1 9-Bit Data Mode

In 9-bit data mode, two bytes of data must be written. The first byte that is loaded into the TX FIFO are the first 8 bits (data bits 7-0) of the 9-bit data. Bit-0 of the second byte that is loaded into the TX FIFO 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.

1.4.2 Receiver

The receiver consists of a 384-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 along with any error tags such as Framing, Parity, Break and Overrun errors. Data from the RX FIFO can be sent to the USB host by sending a bulk-in packet.

1.4.2.1 Character Mode

In character mode, up to 64 bytes of data can be sent at a time to the USB host.

1.4.2.2 Character + Status Mode

In this mode, each 8-bit character and the 4 error bits associated with it can be transmitted to the USB host. The 4 error bits will be in the second byte following the 8-bit character. In this mode, up to 32 character bytes are sent per bulk-in packet.

1.4.2.3 9-Bit Data Mode

In 9-bit data mode, two bytes of data are sent to the USB host for each byte 9-bit data that is received. The first byte sent to the USB host is the first 8-bits of data. Bit-0 of the second byte is the bit-9 of the data.

1.4.3 GPIO

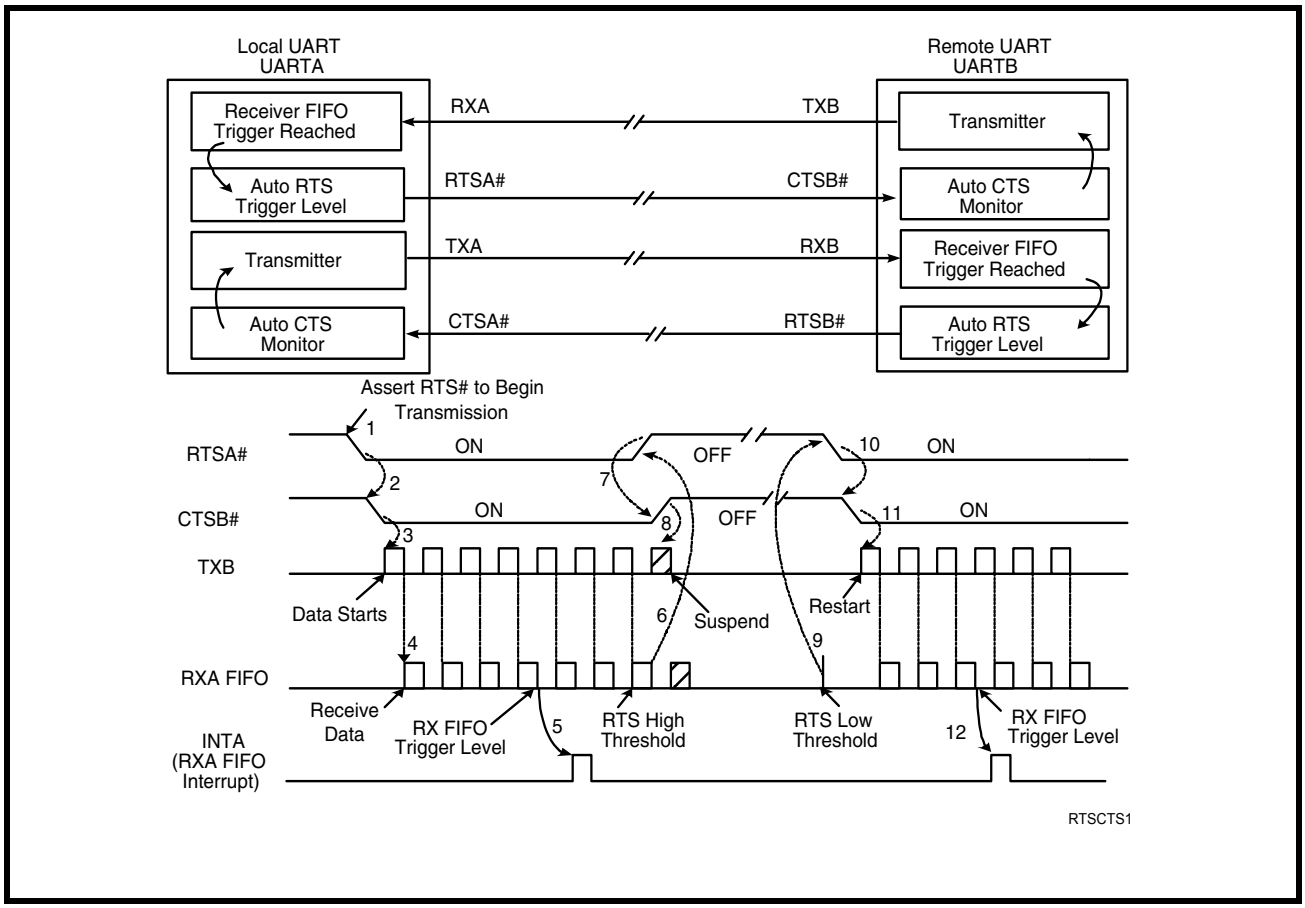
Each UART has 6 GPIOs. By default, the GPIOs are general purpose I/Os. However, there are few modes that can be enabled to add additional feature such as Auto RTS/CTS Flow control, Auto DTR/DSR Flow Control or Transceiver Enable Control. See **Table 13**.

1.4.4 Automatic RTS/CTS Hardware Flow Control

GPIO5 and GPIO4 of the UART channel can be enabled as the RTS# and CTS# signals for Auto RTS/CTS flow control when GPIO_MODE[2:0] = ’001’ and FLOW_CONTROL[2:0] = ’001’. Automatic RTS flow control is used to prevent data overrun errors in local RX FIFO by de-asserting the RTS signal to the remote UART. When there is room in the RX FIFO, the RTS pin will be re-asserted. Automatic CTS flow control is used to

prevent data overrun to the remote RX FIFO. The CTS# input is monitored to suspend/restart the local transmitter (see **Figure 3**):

FIGURE 3. AUTO RTS AND CTS FLOW CONTROL OPERATION



1.4.5 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. For Auto hardware flow control, FLOW_CONTROL[2:0] = '001'. GPIO3 and GPIO2 become DTR# and DSR#, respectively, when GPIO_MODE[2:0] = '010'.

1.4.6 Automatic XON/XOFF Software Flow Control

When software flow control is enabled, the V1412 compares the receive data characters with the programmed Xon or Xoff characters. If the received character matches the programmed Xoff character, the V1412 will halt transmission as soon as the current character has completed transmission. Data transmission is resumed when a received character matches the Xon character. Software flow control is enabled when FLOW_CONTROL[2:0] = '010'.

1.4.7 Multidrop Mode with Automatic Half-Duplex Transceiver Control

Multidrop mode with Automatic Half-Duplex Transceiver Control is enabled when GPIO_MODE[2:0] = '011' and FLOW_CONTROL[2:0] = '011'.

1.4.7.1 Receiver

In this mode, the UART Receiver will automatically be enabled when an address byte (9th bit or parity bit is '1') is received that matches the value stored in the XON_CHAR or XOFF_CHAR register. The address byte will not be loaded into the RX FIFO. 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 register.

1.4.7.2 Transmitter

GPIO5/RTS# pin behaves as control pin for the direction of a half-duplex RS-485 transceiver. The polarity of the GPIO5/RTS# pin can be configured via GPIO_MODE[3]. When the UART is not transmitting data, the GPIO5/RTS# pin will be de-asserted. The GPIO5/RTS# pin will be asserted immediately before the UART starts transmitting data. When the UART is done transmitting data, the GPIO5/RTS# pin will be de-asserted.

1.4.8 Multidrop Mode with Automatic Transmitter Enable

Multidrop mode with Automatic Transmitter Enable is enabled when GPIO_MODE[2:0] = '100' and FLOW_CONTROL[2:0] = '100'.

1.4.8.1 Receiver

The behavior of the receiver is the same in this mode as it is in the Address Match mode described above.

1.4.8.2 Transmitter

When there is an address match with the XON_CHAR register, the GPIO5/RTS# pin is asserted and remains asserted whether the UART is transmitting data or not. The GPIO5/RTS# pin will be de-asserted when an address byte is received that does not match the value in the XON_CHAR register. The polarity of the GPIO5/RTS# pin can be configured via GPIO_MODE[3].

1.4.9 Programmable Turn-Around Delay

By default, the GPIO5/RTS# pin 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 GPIO5/RTS# pin can be delayed from 1 to 15 bit times via the XCVR_EN_DELAY register to allow for the data to reach distant UARTs.

1.4.10 Half-Duplex Mode

Half-duplex mode is enabled when FLOW_CONTROL[3] = 1. In this mode, the UART will ignore any data on the RX input when the UART is transmitting data.

2.0 USB CONTROL COMMANDS

The following table shows all of the USB Control Commands that are supported by the V1412. Commands included are standard USB commands, CDC-ACM commands and custom Exar commands. .

TABLE 3: SUPPORTED USB CONTROL COMMANDS

NAME	REQUEST TYPE	REQUEST	VALUE		INDEX		LENGTH		DESCRIPTION
DEV GET_STATUS	0x80	0	0	0	0	0	2	0	Device: remote wake-up + self-powered
IF GET_STATUS	0x81	0	0	0	1-4, 129-132	0	2	0	Interface: zero
EP GET_STATUS	0x82	0	0	0	0-4, 129-136	0	2	0	Endpoint: halted
DEV CLEAR_FEATURE	0x00	1	1	0	0	0	0	0	Device remote wake-up
EP CLEAR_FEATURE	0x02	1	0	0	0-4, 129-136	0	0	0	Endpoint halt
DEV SET_FEATURE	0x00	3	1	00	0	0	0	0	Device remote wake-up
DEV SET_FEATURE	0x00	3	2	0	0	test	0	0	Test mode
EP SET_FEATURE	0x02	3	0	0	0-4, 129-136	0	0	0	Endpoint halt
SET_ADDRESS	0x00	5	addr	0	0	0	0	0	
GET_DESCRIPTOR	0x80	6	0	1	0	0	len LSB	len MSB	Device descriptor
GET_DESCRIPTOR	0x80	6	0	2	0	0	len LSB	len MSB	Configuration descriptor
GET_CONFIGURATION	0x80	8	0	0	0	0	1	0	
SET_CONFIGURATION	0x00	9	n	0	0	0	0	0	
GET_INTERFACE	0x81	10	0	0	0-7	0	1	0	
CDC_ACM_IF SET_LINE_CODING	0x21	32	0	0	0, 2, 4, 6	0	7	0	Set the UART baud rate, parity, stop bits, etc.
CDC_ACM_IF GET_LINE_CODING	0xA1	33	0	0	0, 2, 4, 6	0	7	0	Get the UART baud rate, parity, stop bits, etc.
CDC_ACM_IF SET_CONTROL_LINE_STATE	0x21	34	val	0	0, 2, 4, 6	0	0	0	Set UART control lines

TABLE 3: SUPPORTED USB CONTROL COMMANDS

NAME	REQUEST TYPE	REQUEST	VALUE		INDEX		LENGTH		DESCRIPTION
CDC_ACM_IF SEND_BREAK	0x21	35	val LSB	val MSB	0, 2, 4, 6	0	0	0	Send a break for the specified duration
XR_SET_REG	0x40	0	val	0	register	block	0	0	Exar custom command: set one 8-bit register val: 8-bit register value register address: see Table 6 block number: see Table 4
XR_GETN_REG	0xC0	1	0	0	register	block	count LSB	count MSB	Exar custom register: get count 8-bit registers register address: see Table 6 block number: see Table 4

2.1 UART Block Numbers

The table below lists the block numbers for accessing each of the UART channels and the UART Manager..

TABLE 4: CONTROL BLOCKS

BLOCK NAME	BLOCK NUMBER	DESCRIPTION
UART Channel A	0	The configuration and control registers for UART channel A.
UART Channel B	1	The configuration and control registers for UART channel B.
Reserved	2-3	Block Numbers 2-3 are reserved.
UART Manager	4	The control registers for the UART Manager. The UART Manager enables/disables the TX and RX FIFOs for each UART.

3.0 REGISTER SET DESCRIPTION

The internal register set of the V1412 consists of 2 different types of registers: UART Manager and UART registers. The UART Manager controls the TX, RX and FIFOs of all UART channels. The UART registers configure and control the remaining UART channel functionality not related to the UART FIFO.

3.1 UART Manager Registers..

TABLE 5: UART MANAGER REGISTERS

ADDRESS	REGISTER NAME	BIT-7	BIT-6	BIT-5	BIT-4	BIT-3	BIT-2	BIT-1	BIT-0
0X10	FIFO_ENABLE_CHA	0	0	0	0	0	0	RX	TX
0X11	FIFO_ENABLE_CHB	0	0	0	0	0	0	RX	TX
0X18	RX_FIFO_RESET_CHA	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0X19	RX_FIFO_RESET_CHB	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x1C	TX_FIFO_RESET_CHA	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x1D	TX_FIFO_RESET_CHB	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0

FIFO_ENABLE Registers

Enables the RX FIFO and TX FIFOs. For proper functionality, the UART TX and RX must be enabled in the following order:

```
FIFO_ENABLE_CHx = 0x1       // Enable TX FIFO
UART_ENABLE = 0x3           // Enable TX and RX of that channel
FIFO_ENABLE_CHx = 0x3       // Enable RX FIFO
```

RX_FIFO_RESET and TX_FIFO_RESET Registers

Writing a non-zero value to these registers resets the FIFOs.

3.2 UART Register Map

TABLE 6: UART REGISTERS

ADDRESS	REGISTER NAME	BIT-7	BIT-6	BIT-5	BIT-4	BIT-3	BIT-2	BIT-1	BIT-0
0X00	Reserved	0	0	0	0	0	0	0	0
0X01	Reserved	0	0	0	0	0	0	0	0
0X02	Reserved	0	0	0	0	0	0	0	0
0X03	UART_ENABLE	0	0	0	0	0	0	RX	TX
0X04	CLOCK_DIVISOR0	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x05	CLOCK_DIVISOR1	Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8
0x06	CLOCK_DIVISOR2	0	0	0	0	0	Bit-18	Bit-17	Bit-16
0x07	TX_CLOCK_MASK0	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x08	TX_CLOCK_MASK1	Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8
0x09	RX_CLOCK_MASK0	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x0A	RX_CLOCK_MASK1	Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8
0x0B	CHARACTER_FORMAT	Stop	Parity			Data Bits			
0x0C	FLOW_CONTROL	0	0	0	0	Half-Duplex	Flow Control Mode Select		
0x0D	Reserved	0	0	0	0	0	0	0	0
0x0E	Reserved	0	0	0	0	0	0	0	0
0x0F	Reserved	0	0	0	0	0	0	0	0
0x10	XON_CHAR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x11	XOFF_CHAR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x12	Reserved	0	0	0	0	0	0	0	0
0x13	ERROR_STATUS	Break Status	Overrun Error	Parity Error	Framing Error	Break Error	0	0	0
0x14	TX_BREAK	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x15	XCVR_EN_DELAY	0	0	0	0	Delay			
0x16	Reserved	0	0	0	0	0	0	0	0
0x17	Reserved	0	0	0	0	0	0	0	0
0x18	Reserved	0	0	0	0	0	0	0	0
0x19	Reserved	0	0	0	0	0	0	0	0
0x1A	GPIO_MODE	0	0	0	0	XCVR Enable Polarity	Mode Select		
0x1B	GPIO_DIRECTION	0	0	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x1D	GPIO_SET	0	0	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x1E	GPIO_CLEAR	0	0	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
0x1F	GPIO_STATUS	0	0	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

3.3 UART Register Descriptions

3.3.1 UART_ENABLE Register Description (Read/Write)

This register enables the UART TX and RX. For proper functionality, the UART TX and RX must be enabled in the following order:

```
FIFO_ENABLE_CHx = 0x1      // Enable TX FIFO
UART_ENABLE = 0x3          // Enable TX and RX of that channel
FIFO_ENABLE_CHx = 0x3      // Enable RX FIFO
```

UART_ENABLE[0]: Enable UART TX

- Logic 0 = UART TX disabled.
- Logic 1 = UART TX enabled.

UART_ENABLE[1]: Enable UART RX

- Logic 0 = UART RX disabled.
- Logic 1 = UART RX enabled.

UART_ENABLE[7:2]: Reserved

These bits are reserved and should remain ’0’.

3.3.2 CLOCK_DIVISOR0, CLOCK_DIVISOR1, CLOCK_DIVISOR2 Register Description (Read/Write)

These registers are used for programming the baud rate. The V1412 uses a 19-bit divisor and 16-bit mask register. Using the internal 48MHz oscillator, the 19-bit divisor is calculated as follows:

```
CLOCK_DIVISOR = Trunc ( 48000000 / Baud Rate )
```

For example, if the the baud rate is 115200bps, then

```
CLOCK_DIVISOR = Trunc ( 48000000 / 115200 ) = Trunc (416.66667) = 416
```

CLOCK_DIVISOR0[7:0]: Baud rate clock divisor bits [7:0]

CLOCK_DIVISOR1[7:0]: Baud rate clock divisor bits [15:8]

CLOCK_DIVISOR2[2:0]: Baud rate clock divisor bits [18:16]

CLOCK_DIVISOR2[7:3]: Reserved

These bits are reserved and should remain ’0’.

3.3.3 TX_CLOCK_MASK0, TX_CLOCK_MASK1 Register Description (Read/Write)

A look-up table is used for the value of the 16-bit TX Clock mask registers. The index of the look-up table is calculated as follows:

```
index = Trunc ( ( ( 48000000  / Baud Rate ) - CLOCK_DIVISOR ) * 32)
```

For example, if the baud rate is 115200bps, then the index will be:

```
index = Trunc ( ( ( 48000000  / 115200 ) - 416 ) * 32) = Trunc (21.3333) = 21
```

The values for some baud rates to program the TX_CLOCK_MASK registers are listed in Table 7. For baud rates that are not listed, use the index to select TX_CLOCK_MASK register values from Table 8.

3.3.4 RX_CLOCK_MASK0, RX_CLOCK_MASK1 Register Description (Read/Write)

The values for some baud rates to program the RX_CLOCK_MASK registers are listed in Table 7. For baud rates that are not listed, use the same index calculated for the TX_CLOCK_MASK register to select RX_CLOCK_MASK register values from Table 8.

TABLE 7: CLOCK DIVISOR AND CLOCK MASK VALUES FOR COMMON BAUD RATES

BAUD RATE (BPS)	CLOCK DIVISOR (DECIMAL)	TX CLOCK MASK (HEX)	RX CLOCK MASK (HEX)
1200	40000	0x0000	0x0000
2400	20000	0x0000	0x0000
4800	10000	0x0000	0x0000
9600	5000	0x0000	0x0000
19200	2500	0x0000	0x0000
38400	1250	0x0000	0x0000
57600	833	0x0912	0x0924
115200	416	0x0B6D	0x0B6A
230400	208	0x0912	0x0924
460800	104	0x0208	0x0040
500000	96	0x0000	0x0000
576000	83	0x0912	0x0924
921600	52	0x0040	0x0000
1000000	48	0x0000	0x0000
1152000	41	0x0B6D	0x0DB6
1500000	32	0x0000	0x0000
2000000	24	0x0000	0x0000
2500000	19	0x0104	0x0108
3000000	16	0x0000	0x0000
3125000	15	0x0492	0x0492
3500000	13	0x076D	0x0BB6
4000000	12	0x0000	0x0000
4250000	11	0x0122	0x0224
6250000	7	0x0B6D	0x0DB6
8000000	6	0x0000	0x0000
12000000	4	0x0000	0x0000

For baud rates that are not listed in the table above, use the index value calcuated using the formula in **"Section 3.3.3, TX_CLOCK_MASK0, TX_CLOCK_MASK1 Register Description (Read/Write)" on page 15** to determine which TX Clock and RX Clock Mask register values to use from **Table 8**. For the the RX Clock Mask register, there are 2 values listed and would depend on whether the Clock Divisor is even or odd. For even Clock Divisors, use the value from the first column. For odd Clock Divisors, use the value from the last column.

TABLE 8: TX AND RX CLOCK MASK VALUES

INDEX (DECIMAL)	TX CLOCK MASK (HEX)	RX CLOCK MASK (HEX) - EVEN CLOCK DIVISOR	RX CLOCK MASK (HEX) - ODD CLOCK DIVISOR
0	0x0000	0x0000	0x0000
1	0x0000	0x0000	0x0000
2	0x0100	0x0000	0x0100
3	0x0020	0x0400	0x0020
4	0x0010	0x0100	0x0010
5	0x0208	0x0040	0x0208
6	0x0104	0x0820	0x0108
7	0x0844	0x0210	0x0884
8	0x0444	0x0110	0x0444
9	0x0122	0x0888	0x0224
10	0x0912	0x0448	0x0924
11	0x0492	0x0248	0x0492
12	0x0252	0x0928	0x0292
13	0x094A	0x04A4	0x0A52
14	0x052A	0x0AA4	0x054A
15	0x0AAA	0x0954	0x04AA
16	0x0AAA	0x0554	0x0AAA
17	0x0555	0x0AD4	0x05AA
18	0x0B55	0x0AB4	0x055A
19	0x06B5	0x05AC	0x0B56
20	0x05B5	0x0D6C	0x06D6
21	0x0B6D	0x0B6A	0x0DB6
22	0x076D	0x06DA	0x0BB6
23	0x0EDD	0x0DDA	0x076E
24	0x0DDD	0x0BBA	0x0EEE
25	0x07BB	0x0F7A	0x0DDE
26	0x0F7B	0x0EF6	0x07DE
27	0x0DF7	0x0BF6	0x0F7E
28	0x07F7	0x0FEE	0x0EFE
29	0x0FDF	0x0FBE	0x07FE
30	0x0F7F	0x0EFE	0x0FFE
31	0x0FFF	0x0FFE	0x0FFD

3.3.5 CHARACTER_FORMAT Register Description (Read/Write)

This register controls the character format such as the word length (7, 8 or 9), parity (odd, even, forced '0', or forced '1') and number of stop bits (1 or 2).

CHARACTER_FORMAT[3:0]: Data Bits.

TABLE 9: DATA BITS

DATA BITS	CHARACTER_FORMAT[3:0]
7	0111
8	1000
9	1001

All other values for CHARACTER_FORMAT[3:0] are reserved.

CHARACTER_FORMAT[6:4]: Parity Mode Select

These bits select the parity mode. If 9-bit data mode has been selected, then writing to these bits will not have any effect. In other words, there will not be an additional parity bit.

TABLE 10: PARITY SELECTION

BIT-6	BIT-5	BIT-4	PARITY SELECTION
0	0	0	No parity
0	0	1	Odd parity
0	1	0	Even parity
0	1	1	Force parity to mark, "1"
1	0	0	Force parity to space, "0"

CHARACTER_FORMAT[7]: Stop Bit select

This register selects the number of stop bits to add to the transmitted character and how many stop bits to check for in the received character.

TABLE 11: STOP BIT SELECTION

BIT-7	NUMBER OF STOP BITS
0	1 stop bit
1	2 stop bits

3.3.6 FLOW_CONTROL Register Description (Read/Write)

These registers select the flow control mode. These registers 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.

FLOW_CONTROL[2:0]: Flow control mode select

TABLE 12: FLOW CONTROL MODE SELECTION

BIT-2	BIT-1	BIT-0	MODE DESCRIPTION
0	0	0	No flow control.
0	0	1	HW flow control enabled
0	1	0	SW flow control enabled
0	1	1	Multidrop mode with Automatic Half-Duplex Transceiver control
1	0	0	Multidrop mode with Automatic Transmitter Enable

FLOW_CONTROL[3]: Half-Duplex Mode

- Logic 0 = Normal (full-duplex) mode. The UART can transmit and receive data at the same time.
- Logic 1 = Half-duplex Mode. In half-duplex mode, any data on the RX pin is ignored when the UART is transmitting data.

FLOW_CONTROL[7:4]: Reserved

These bits are reserved and should remain '0'.

3.3.7 XON_CHAR, XOFF_CHAR Register Descriptions (Read/Write)

The XON_CHAR and XOFF_CHAR registers store the XON and XOFF characters, respectively, that are used in the Automatic Software Flow control.

XON_CHAR[7:0]: XON Character

In Automatic Software Flow control mode, the UART will resume data transmission when the XON character has been received.

For behavior in the Address Match mode, see **"Section 1.4.7, Multidrop Mode with Automatic Half-Duplex Transceiver Control" on page 10**.

For behavior in the Address Match with TX Flow Control mode, see **"Section 1.4.8, Multidrop Mode with Automatic Transmitter Enable" on page 10**.

XOFF_CHAR[7:0]: 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 **"Section 1.4.7, Multidrop Mode with Automatic Half-Duplex Transceiver Control" on page 10**.

For behavior in the Address Match with TX Flow Control mode, see **"Section 1.4.8, Multidrop Mode with Automatic Transmitter Enable" on page 10**.

3.3.8 ERROR_STATUS Register Description - Read-only

This register reports any errors that may have occurred on the line such as break, framing, parity and overrun.

ERROR_STATUS[2:0]: Reserved

These bits are reserved. Any values read from these bits should be ignored.

ERROR_STATUS[3]: Break error

- Logic 0 = No break condition
- Logic 1 = A break condition has been detected (clears after read).

ERROR_STATUS[4]: Framing Error

- Logic 0 = No framing error
- Logic 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.

ERROR_STATUS[5]: Parity Error

- Logic 0 = No parity error
- Logic 1 = A parity error has been detected (clears after read).

ERROR_STATUS[6]: Overrun Error

- Logic 0 = No overrun error
- Logic 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.

ERROR_STATUS[7]: Break Status

- Logic 0 = Break condition is no longer present.
- Logic 1 = Break condition is currently being detected.

3.3.9 TX_BREAK Register Description (Read/Write)

Writing a non-zero value to this register causes a break condition to be generated continuously until the register is cleared. If data is being shifted out of the TX pin, the data will be completed shifted out before the break condition is generated.

3.3.10 XCVR_EN_DELAY Register Description (Read/Write)

XCVR_EN_DELAY[3:0]: Turn-around delay

This is the number of bit times to wait before changing the direction of the transceiver from transmit to receive when half-duplex mode is enabled.

XCVR_EN_DELAY[3:0]: Reserved

These bits are reserved and should be ’0’.

3.3.11 GPIO_MODE Register Description (Read/Write)

GPIO_MODE[2:0]: GPIO Mode Select

There are 4 modes of operation for the GPIOs. The descriptions can be found in **“Section 1.4, UART” on page 7**.

TABLE 13: GPIO MODES

BITS [2:0]	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	MODE DESCRIPTION
000	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	GPIO Mode
001	GPIO0	GPIO1	GPIO2	GPIO3	CTS#	RTS#	Auto RTS/CTS HW Flow Control
010	GPIO0	GPIO1	DSR#	DTR#	GPIO4	GPIO5	Auto DTR/DSR HW Flow Control
011	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	XCVR Enable	Multidrop Mode with Auto Half-Duplex Transceiver Control
100	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	XCVR Enable	Multidrop Mode with Auto TX Enable

GPIO_MODE[3]: Transceiver Enable Polarity

- Logic 0 = Low for TX
- Logic 1 = High for TX

GPIO_MODE[7:4]: Reserved

These register bits are reserved. When writing to these bits, the value should be ’0’. When reading from these bits, they are undefined and should be ignored.

3.3.12 GPIO_DIRECTION Register Description (Read/Write)

This register controls the direction of the GPIO if it is not controlled by the GPIO_MODE register.

GPIO_DIRECTION[5:0]: GPIOx Direction

- Logic 0 = GPIOx is an input.
- Logic 1 = GPIOx is an output.

GPIO_DIRECTION[7:6]: Reserved

These register bits are reserved and should be ’0’.

3.3.13 GPIO_SET Register Description (Read/Write)

Writing a ’1’ in this register drives the GPIO output high. Writing a ’0’ to a bit has no effect. Bits 7-6 are unused and should be ’0’.

3.3.14 GPIO_CLEAR Register Description (Read/Write)

Writing a ’1’ in this register drives the GPIO output low. Writing a ’0’ to a bit has no effect. Bits 7-6 are unused and should be ’0’.

3.3.15 GPIO_STATUS Register Description (Read-Only)

This register reports the current state of the GPIO pin.

4.0 ELECTRICAL CHARACTERISTICS

DC ELECTRICAL CHARACTERISTICS - POWER CONSUMPTION

UNLESS OTHERWISE NOTED: TA = -40° TO +85°C, VCC IS 2.97 TO 3.63V

SYMBOL	PARAMETER	LIMITS 3.3V MIN	TYP	MAX	UNITS	CONDITIONS
I_{CC}	Power Supply Current		16	20	mA	
I_{Susp}	Suspend mode Current		2	2.15	mA	

DC ELECTRICAL CHARACTERISTICS - UART & GPIO PINS

UNLESS OTHERWISE NOTED: TA = -40° TO +85°C, VCC IS 2.97 TO 3.63V

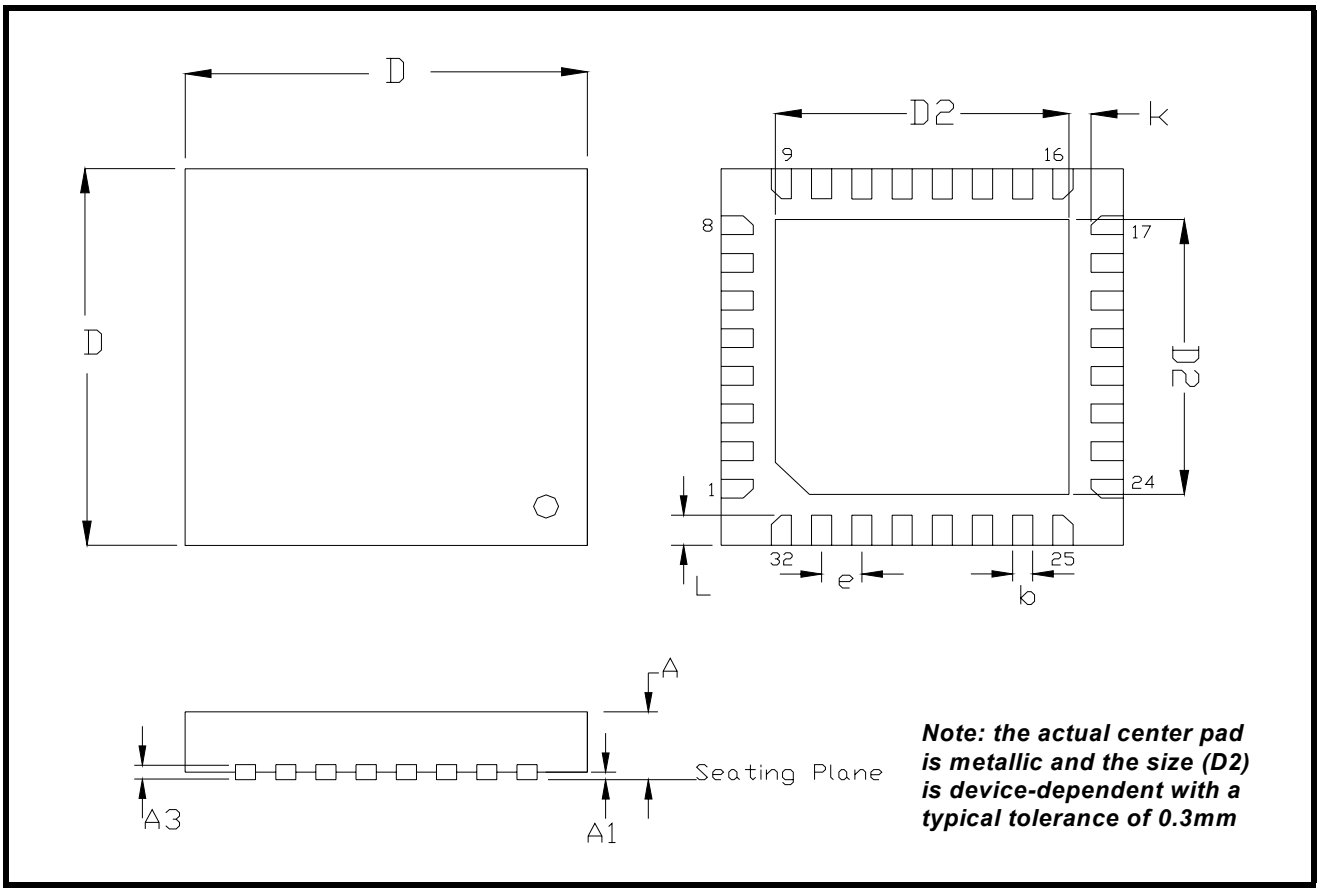
SYMBOL	PARAMETER	LIMITS 3.3V MIN	MAX	UNITS	CONDITIONS
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	5.5	V	
V_{OL}	Output Low Voltage		0.3	V	I_{OL} = 4 mA
V_{OH}	Output High Voltage	2.2		V	I_{OH} = -4 mA
I_{IL}	Input Low Leakage Current		±10	uA	
I_{IH}	Input High Leakage Current		±10	uA	
C_{IN}	Input Pin Capacitance		5	pF	

DC ELECTRICAL CHARACTERISTICS - USB I/O PINS

UNLESS OTHERWISE NOTED: TA = -40° TO +85°C, VCC IS 2.97 TO 3.63V

SYMBOL	PARAMETER	LIMITS 3.3V MIN	MAX	UNITS	CONDITIONS
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	0	0.3	V	External 15 K Ohm to GND on USBD- pin
V_{OH}	Output High Voltage	2.8	3.6	V	External 15 K Ohm to GND on USBD- pin
V_{DrvZ}	Driver Output Impedance	28	44	Ohms	
I_{OSC}	Open short current Current		35	mA	1.5 V on USBD+ and USBD-

PACKAGE DIMENSIONS (32 PIN QFN - 5 X 5 X 0.9 mm)



Note: The control dimension is in millimeter.

SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.031	0.039	0.80	1.00
A1	0.000	0.002	0.00	0.05
A3	0.006	0.010	0.15	0.25
D	0.193	0.201	4.90	5.10
D2	0.138	0.150	3.50	3.80
b	0.007	0.012	0.18	0.30
e	0.0197 BSC		0.50 BSC	
L	0.012	0.020	0.35	0.45
k	0.008	-	0.20	-

REVISION HISTORY

DATE	REVISION	DESCRIPTION
June 2009	1.0.0	Final Datasheet.

NOTICE

Datasheet June 2009.

Send your UART technical inquiry with technical details to hotline: uarttechsupport@exar.com.