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 XR21B1411 (B1411) is an enhanced 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 B1411 operates from an internal clock that is programmable to 6, 12, 24 or 48 MHz. Therefore, no external crystal / oscillator is required as in previous generation UARTs. With the fractional baud rate generator, any baud rate up to 12 Mbps can be accurately generated using the internal clock.

The large 128-byte Tx FIFO and 384-byte Rx FIFO of the B1411 help 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 Vendor ID, Product ID, bus-powered mode, self-powered mode, remote wakeup support or maximum power consumption values, as well as default baud rate settings can be programmed using the on-board OTP through the USBD+ / USBD- pins.

The B1411 operates from a single 5V power supply and is available in a 16-pin QFN package.

WHQL certified software drivers for Windows 2000, XP, Vista, 7, 8, 8.1, 10 and CE, as well as Linux and Mac are supported for the XR21B1411.

APPLICATIONS

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

FEATURES

- ±15kV HBM ESD on USBD+/USBD-
- USB 2.0 Compliant, Full-Speed (12 Mbps)
  - Supports USB suspend, resume and remote wakeup operations
- Unique preprogrammed USB serial number
- Enhanced UART Features
  - 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
  - Odd, even, mark, space or no parity
  - Automatic Hardware (RTS/CTS or DTR/DSR) Flow Control
  - Automatic Software (Xon/Xoff) Flow Control
  - Multidrop mode
  - Auto RS-485 Half-Duplex Control
  - Half-Duplex mode
  - Selectable GPIO or Modem I/O
- Internal 48 MHz clock with clock divisors programmable down to 6 MHz
- Single 5V power supply
- 5V tolerant inputs
- 16-pin QFN package
- Virtual COM Port WHQL/HCK Windows certified drivers
  - Windows 2000, XP, Vista, 7, 8, 8.1 and 10
  - Windows CE 4.2, 5.0, 6.0, 7.0
- Linux, Mac drivers

FIGURE 1. XR21B1411 BLOCK DIAGRAM

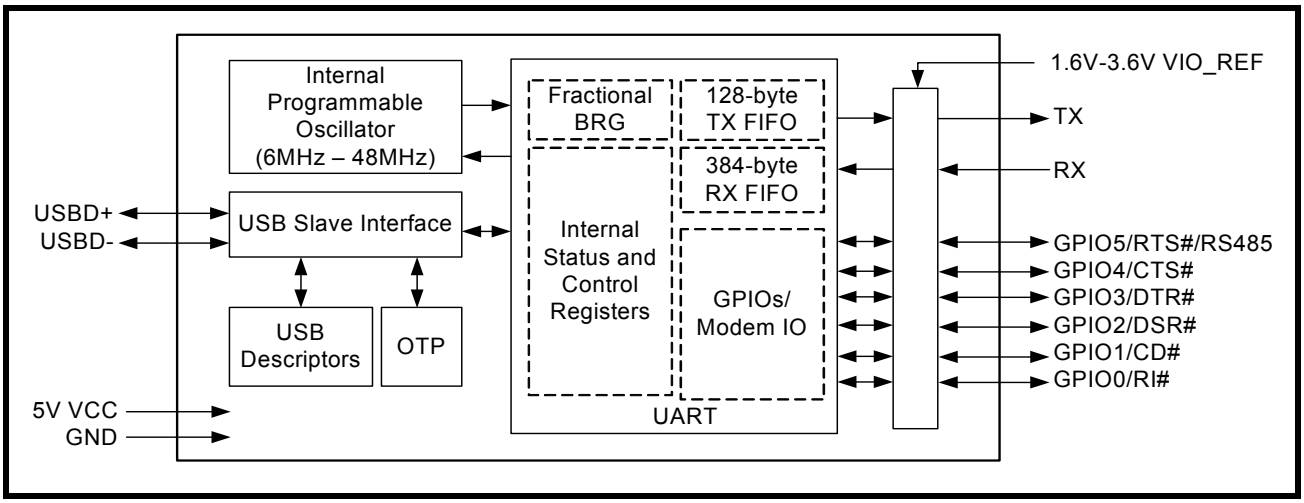
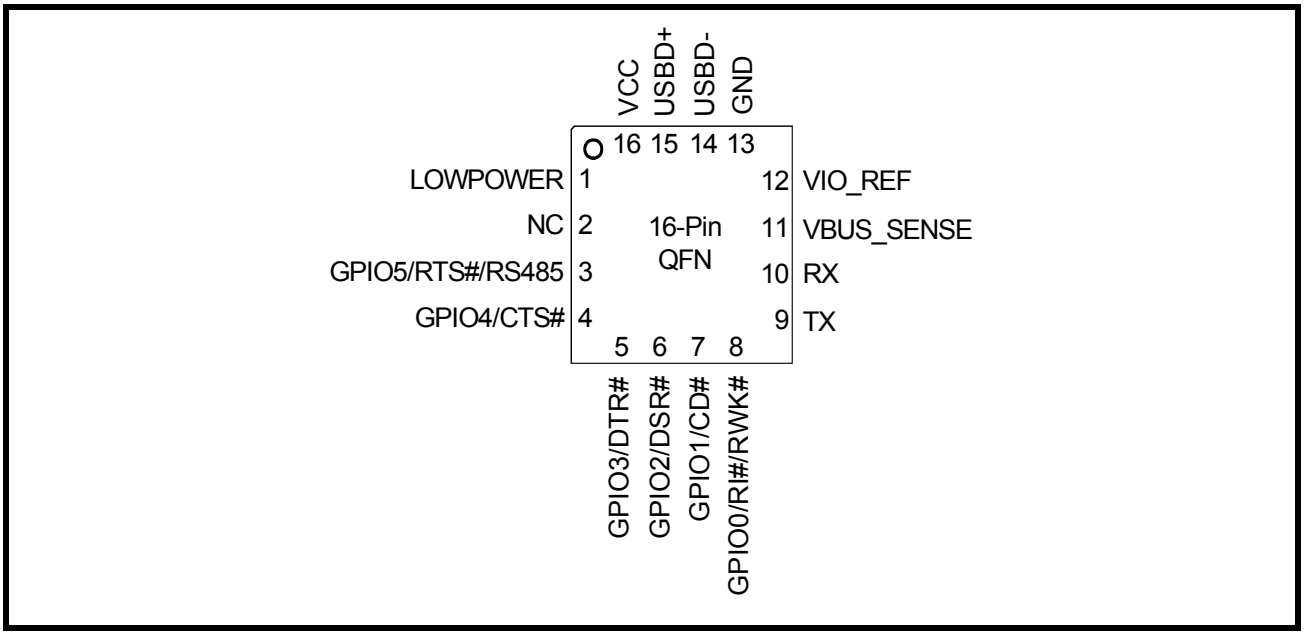


FIGURE 2. PIN OUT DIAGRAM



ORDERING INFORMATION

PART NUMBER	PACKAGE	OPERATING TEMPERATURE RANGE	DEVICE STATUS
XR21B1411IL16-F	16-pin QFN	-40°C to +85°C	Active
XR21B1411IL16TR-F	16-pin QFN	-40°C to +85°C	Active
XR21B1411IL16MTR-F	16-pin QFN	-40°C to +85°C	Active

NOTE: **TR = Tape and Reel, MTR = Mini Tape and Reel, F = Green / RoHS**

PIN DESCRIPTIONS

Pin Description

NAME	16-QFN PIN #	TYPE	DESCRIPTION
UART Signals			
RX	10	I	UART Receive Data. This pin has a programmable pull-up or pull-down resistor which may be enabled by OTP programming. Pull-up resistor will be disabled without valid voltage on VIO_REF pin.
TX	9	O	UART Transmit Data.
GPIO0/RI#/RWK#	8	I/O	General purpose I/O or UART Ring-Indicator input (active low) or Remote Wakeup input. This pin has a programmable pull-up or pull-down resistor which may be enabled by OTP programming. This pin may also be used by any device to signal the USB host to exit the Suspend state. **See "Section 1.3.10, USB Suspend" on page 9.**
GPIO1/CD#	7	I/O	General purpose I/O or UART Carrier-Detect input (active low). This pin has a programmable pull-up or pull-down resistor which may be enabled by OTP programming.
GPIO2/DSR#	6	I/O	General purpose I/O or UART Data-Set-Ready input (active low). **See "Section 1.3.5.2, Automatic DTR/DSR Hardware Flow Control" on page 8.** This pin has a programmable pull-up or pull-down resistor which may be enabled by OTP programming.
GPIO3/DTR#	5	I/O	General purpose I/O or UART Data-Terminal-Ready output (active low). **See "Section 1.3.5.2, Automatic DTR/DSR Hardware Flow Control" on page 8.** This pin has a programmable pull-up or pull-down resistor which may be enabled by OTP programming. This bit will be automatically configured as an output when using the standard CDC-ACM driver.
GPIO4/CTS#	4	I/O	General purpose I/O or UART Clear-to-Send input (active low). **See "Section 1.3.5.1, Automatic RTS/CTS Hardware Flow Control" on page 8.** This pin has a programmable pull-up or pull-down resistor which may be enabled by OTP programming.
GPIO5/RTS#/RS485	3	I/O	General purpose I/O or UART Request-to-Send output (active low) or auto RS-485 half-duplex control. See **"Section 1.3.5.1, Automatic RTS/CTS Hardware Flow Control" on page 8** or **"Section 1.3.6, Auto RS-485 Half-Duplex Control" on page 8**. This pin has a programmable pull-up or pull-down resistor which may be enabled by OTP programming. This bit will be automatically configured for hardware flow control as RTS# output when using the standard CDC-ACM driver.
USB Interface Signals			
USBD+	15	I/O	USB port differential data positive input. This pin has internal pull-up resistor compliant to USB 2.0 specification. The ESD protection on this pin is +/-15 kV HBM.
USBD-	14	I/O	USB port differential data negative input. The ESD protection on this pin is +/-15 kV HBM.

Pin Description

NAME	16-QFN PIN #	TYPE	DESCRIPTION
Miscellaneous Signals			
LOWPOWER	1	O	The LOWPOWER pin will be asserted whenever it is not safe to draw the amount of current requested from VBUS in the Device Maximum Power field of the Configuration Descriptor. The LOWPOWER pin will behave differently for a low power device and a high power device. ■ Low-power device (<= 1 unit load or 100 mA i.e. bMaxPower <= 0x32): LOWPOWER pin is asserted when the USB UART is in suspend mode. ■ High-power deivce (bMaxPower > 0x32): LOWPOWER pin is asserted when the USB UART is in suspend mode or when it is not yet configured. The LOWPOWER pin will be de-asserted whenever it is safe to draw the amount of current from VBUS requested in the Device Maximum Power field. The default active low polarity may be changed via the OTP. Connect this pin to VIO_REF or to ground through a weak (10K) pull-up or pull-down resistor to match the polarity of the asserted state in high power applications. In low power applications, no external resistor is required.
VBUS_SENSE	11	I	VBUS Sense input. This pin is used to disable the pull-up resistor on the USBD+ signal when VBUS is not present in self-powered mode. In self-powered mode, the VBUS from the USB connector should be connected to this pin through a voltage divider circuit (VBUS = 5V), such that VBUS_SENSE = VIO_REF, using large resistance values to minimize power. It should also be decoupled by a 0.1 uF capacitor. This feature must be enabled via the OTP. In bus-powered mode, this pin is ignored but should be tied to a defined logic state.
NC	2		No Connect
Power / Ground Signals			
VIO_REF	12	Pwr	Reference voltage for the modem I/O signals. The voltage range for VIO_REF is + 1.6V to + 3.6V.
VCC	16	Pwr	5V power supply. Input voltage range for VCC is + 4.4V to + 5.25V.
GND	13	Pwr	Power supply common, ground.
GND	Center Pad	Pwr	The center pad on the back side of the QFN package is metallic and should be connected to GND on the PCB. The thermal pad size on the PCB should be the approximate size of this center pad and should be solder mask defined. The solder mask opening should be at least 0.0025" inwards from the edge of the PCB thermal pad.

NOTES:

1. *Pin type: I=Input, O=Output, I/O= Input/output, PWR=Power, OD=Output Open Drain.*
2. *IO pins are undefined during USB bus reset and POR (Power On Reset). To ensure defined state during reset conditions, use a weak external resistor to pull to the desired state.*

1.0 FUNCTIONAL DESCRIPTIONS

1.1 USB interface

The USB interface of the B1411 is compliant with the USB 2.0 Full-Speed Specifications.

The B1411 uses the following set of parameters:

- 1 Control Endpoint
  - Endpoint 0 as outlined in the USB specifications
- 1 Configuration is supported
- 2 interfaces for the UART channel
  - Single interrupt endpoint
  - 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 B1411. This value can be changed by programming the internal OTP via the USB link.

1.1.2 USB Product ID

The default USB Product ID for the B1411 is 0x1411. This value can be changed by programming the internal OTP via the USB link. Note that Exar's custom drivers for all Windows OS require that the Product ID be an odd number for the B1411 device for proper identification of the device.

1.2 USB Device Driver

The B1411 device can be used with either a standard CDC-ACM driver or a custom driver. When the CDC-ACM driver is used, the driver has no capability to read or write the B1411 device registers. Because of this, the B1411 device is initialized to the settings in Table 1. With a custom driver, all GPIOs default in hardware to inputs but these settings may be modified by the custom driver.

TABLE 1: B1411 REGISTER DEFAULTS WITH CDC-ACM DRIVER

REGISTER	VALUE	NOTES
FLOW_CONTROL	0x001	Hardware flow control
GPIO_MODE	0x001	RTS / CTS flow control
GPIO_DIRECTION	0x008	GPIO3/DTR# configured as an output
GPIO_INT_MASK	0x030	GPIO0/RI#, GPIO1/CD# and GPIO2/DSR# are interrupt sensitive, i.e. can cause a USB interrupt to be generated

These default settings can be overridden by programming the OTP via the Address Value feature.

Although there is no ability to read / write registers when using the CDC-ACM driver, basic UART functions, including setting baud rate, character format and sending line break is supported by the CDC driver. Refer to the 4 CDC_ACM_IF USB Control Commands listed in Table 2.

If a custom driver is used, the CUSTOM_DRIVER_ACTIVE bit should be immediately set to '1' by the driver. Once the CUSTOM_DRIVER_ACTIVE bit is set, the custom driver can use standard CDC-ACM commands without configuring the device to the default register settings used with the CDC-ACM driver. Any changes to the register settings for the GPIOs and flow control will specifically need to be configured by the driver.

1.3 UART

The UART can be configured via USB control transfers from the USB host. The UART transmitter and receiver sections are described separately below.

1.3.1 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. 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 can be configured for 7 or 8 data bits with or without parity or 9 data bits without parity.

If 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 always be a '0' unless the wide mode is enabled.

1.3.1.1 Wide mode Transmit

When 9 bit data and the wide mode are both selected, 2 bytes from the USB host are used to form 9 bit data which is serialized and transmitted on the UART TX pin. The first byte received into the TX FIFO forms the first 8 bits of data and the least significant bit of the second byte forms the 9th data bit. The remaining 7 bits of the second byte are discarded. The wide mode can be enabled via the WIDE_MODE register at address 0xD02.

1.3.2 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. Data from the RX FIFO is sent to the USB host 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.

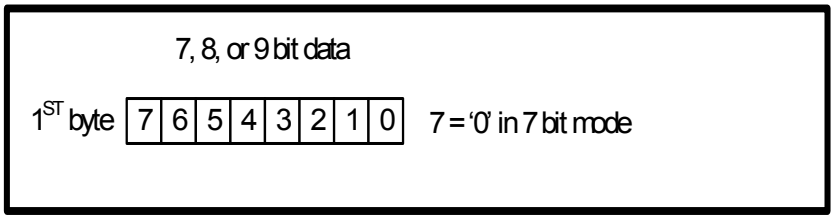
1.3.2.1 Normal receive operation with 7 or 8-bit data

Data that is received is stored in the RX FIFO. Any parity, framing or overrun error or break status information related to the data is discarded. Receive data format is shown in Figure 3.

1.3.2.2 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.

FIGURE 3. NORMAL OPERATION RECEIVE DATA FORMAT



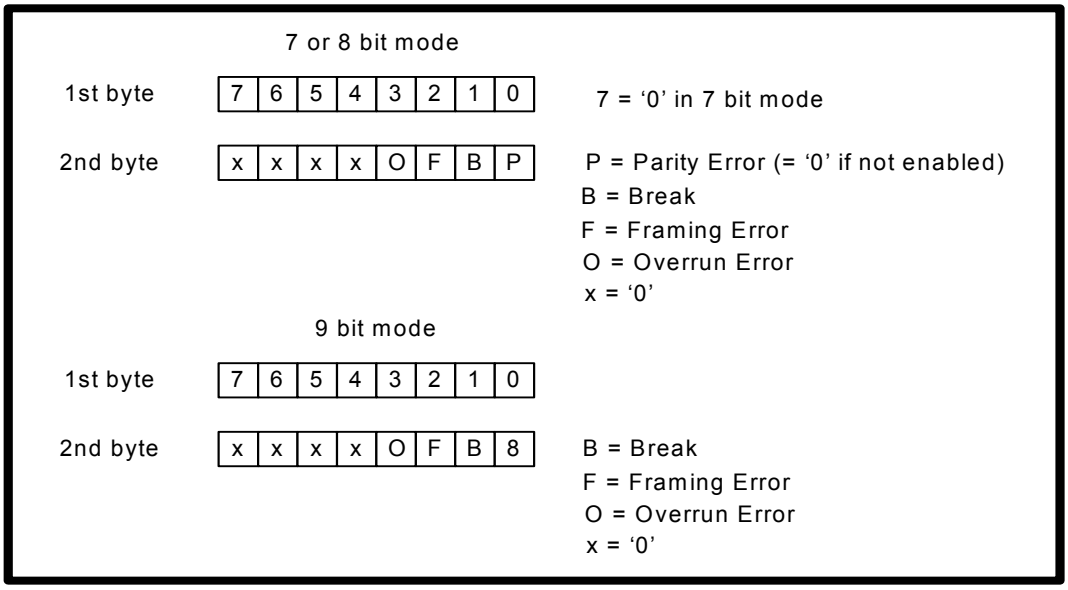
1.3.2.3 Wide mode operation with 7 or 8-bit data

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 4.

1.3.2.4 Wide mode 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.

FIGURE 4. WIDE MODE RECEIVE DATA FORMAT



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 in this manner.

1.3.3 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 B1411 will acknowledge a bulk-in request from the host and transfer the data packet. If there is less than 64 bytes in the RX FIFO, the B1411 may NAK the bulk-in request indicating that data is not ready to transfer at that time. However, if there is less than 64 bytes in the RX FIFO and no data has been received for more than 3 character times, the B1411 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 increases latency unacceptably. The B1411 has a low latency register bit that will cause the B1411 to immediately transfer any received data in the RX FIFO to the USB host, i.e. it will not wait for 3 character times. The custom driver can automatically set the RX_FIFO_LOW_LATENCY register bit to force the B1411 to be in the low latency mode, or the user may manually set this bit. With the CDC-ACM driver, the low latency mode is automatically set whenever the baud rate is set to a value of less than 40961 bps using the CDC_ACM_IF_SET_LINE_CODING command.

1.3.4 GPIO

The UART has 6 GPIOs. By hardware default the GPIOs are configured as inputs but may be modified by a custom driver. However, they can also be configured to add additional features such as Auto RTS/CTS Flow control, Auto DTR/DSR Flow Control or auto RS-485 half duplex control. Both GPIO modes 3 and 4 may be used to automatically assert GPIO5 as auto RS-485 half duplex control. See **Table 7** for the register control and details of GPIO modes. Note that settings in the GPIO mode register should coordinate with settings in the Flow Control mode register described in **"Section 1.3.5, Flow Control" on page 7**. Not all combinations of these two registers will be valid. See **"Section 1.3.7, Multidrop mode with address matching" on page 9** for more details regarding Rx address matching.

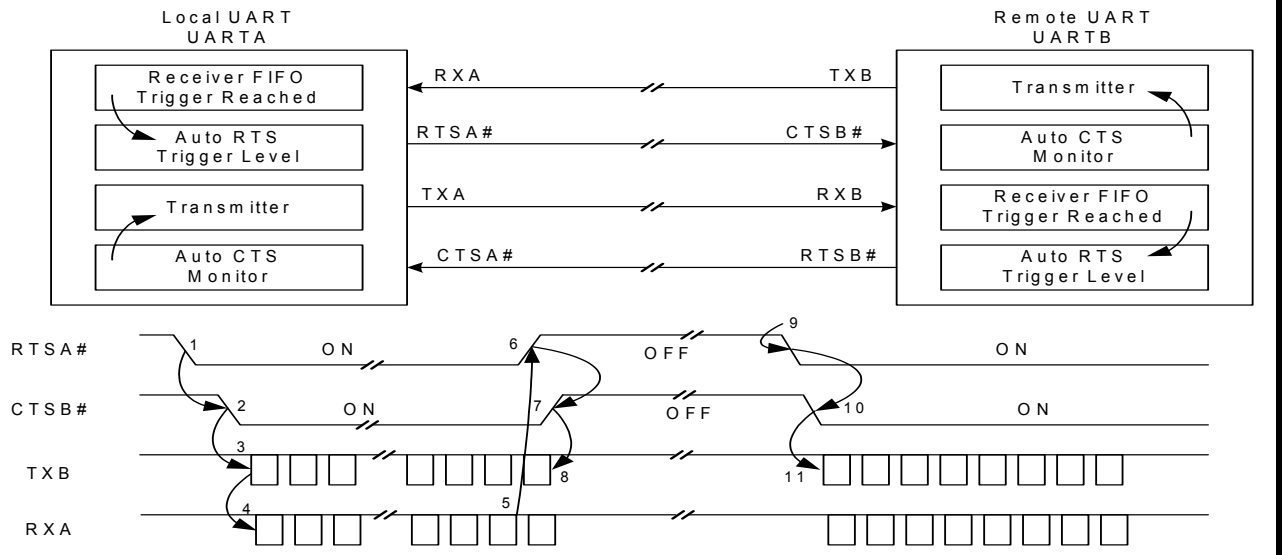
1.3.5 Flow Control

The B1411 can perform both hardware and software flow control. Software flow control is selected by flow control mode 2. Hardware flow control can either be RTS/CTS or DTR/DSR controlled and is selected by flow control mode 1. See **Table 6** for the register control and details of flow control modes. The following sections describe the three types of flow control which may be used.

1.3.5.1 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'. 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 (refer to **Figure 5**):

FIGURE 5. AUTO RTS AND CTS FLOW CONTROL OPERATION



1.3.5.2 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. GPIO3 and GPIO2 become DTR# and DSR#, respectively, when GPIO_MODE[2:0] = '010' and FLOW_CONTROL[2:0] = '001'.

1.3.5.3 Automatic XON/XOFF Software Flow Control

When software flow control is enabled, the B1411 compares the receive data characters with the programmed Xon or Xoff characters. If the received character matches the programmed Xoff character, the B1411 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'.

1.3.6 Auto RS-485 Half-Duplex Control

The Auto RS-485 Half-Duplex Control feature changes the behavior of the GPIO5/RTS#/RS485 pin when enabled by the GPIO_MODE register bits 2-0. **See "Section 3.1.1.13, GPIO_MODE Register Description (Read / Write)" on page 18.** The FLOW_CONTROL register must also be set appropriately for use in

multidrop applications. See "Section 3.1.1.7, FLOW_CONTROL Register Description (Read / Write)" on page 15. If enabled, the transmitter automatically asserts the GPIO5/RTS#/RS485 output prior to sending the data. By default, it de-asserts GPIO5/RTS#/RS485 following the last stop bit of the last character that has been transmitted, but the RS485_DELAY register may be used to delay the deassertion. The polarity of the GPIO5/RTS#/RS485 signal may also be modified using the GPIO_MODE register bit 3.

1.3.7 Multidrop mode with address matching

The B1411 device has two address matching modes which are also set by the flow mode control register using modes 3 and 4. These modes are intended for a multi-drop network application. In these modes, the XON_CHAR register holds a unicast address and the XOFF_CHAR holds a multicast address. A unicast address is used by a transmitting master to broadcast an address to all attached slave devices that is intended for only one slave device. A multicast address is used to broadcast an address intended for more than one recipient device. Each attached slave device should have a unique unicast address value stored in the XON_CHAR register, while multiple slaves may have the same multicast adderss stored in the XOFF_CHAR register. An address match occurs when an address byte (9th bit or parity bit is '1') is received that matches the value stored in either the XON_CHAR or XOFF_CHAR register.

1.3.7.1 Receiver

If an address match occurs in either flow control mode 3 or 4, the address byte will not be loaded into the RX FIFO, but 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.

1.3.7.2 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.

1.3.8 Programmable Turn-Around Delay

By default, the GPIO5/RTS#/RS485 pin will be de-asserted immediately after the stop bit of the last byte has been shifted when auto RS-485 half-duplex control is enabled by the GPIO_MODE register. However, this may not be ideal for systems where the signal needs to propagate over long cables. Therefore, the de-assertion of GPIO5/RTS#/RS485 pin can be delayed from 1 to 15 bit times via the RS485_DELAY register to allow for the data to reach distant UARTs.

1.3.9 Half-Duplex Mode

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

1.3.10 USB Suspend

All USB peripheral devices must support the USB suspend mode. Per USB standard, the B1411 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 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 B1411) that draws power from the host VBUS.

1.3.11 Remote Wakeup

When the B1411 is suspended, the GPIO0/RI#/RWK# pin can be used to request that the host exit the Suspend state. A high to low transition on this pin will cause the device to signal a remote wakeup request to the host via a custom driver. Note that the standard CDC-ACM driver does not support this feature.

In order for the remote wakeup to work, several things must be properly configured. First, the GPIO0/RI# pin must be configured as an input. Additionally, the B1411 device must have the remote wakeup feature support indicated in the USB attributes - See "Section 3.2.1.11, USB_ATTRIBUTES (Read / Write OTP)" on page 25. Lastly, the host must detect the B1411 support for remote wake up and enable this feature. Note that per USB standard, any remote wakeup signaling to the host will be suppressed for the first 5 ms after the device enters the suspend state.

1.4 OTP

The OTP is an on-chip non-volatile memory, that is 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 addersses given in this data sheet.

The OTP memory contains user programmable locations for customer vendor and product ID and device attributes. **Table 11** lists all of the OTP memory contents.

2.0 USB CONTROL COMMANDS

The following table shows all of the USB Control Commands that are supported by the B1411. Commands include standard USB commands, CDC-ACM commands and custom Exar commands. .

TABLE 2: 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	0, 1	0	2	0	Interface: zero
EP GET_STATUS	0x82	0	0	0	0,1, 129, 133	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	0,1, 129, 133	0	0	0	Endpoint halt
DEV SET_FEATURE	0x00	3	1	0	0	0	0	0	Device remote wake-up
DEV SET_FEATURE	0x00	3	2	0	0	test	0	0	Test mode - factory use only
EP SET_FEATURE	0x02	3	0	0	1, 129, 133	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_DESCRIPTOR	0x80	6	0	3	0	0	len LSB	len MSB	String descriptor
GET_CONFIGURATION	0x80	8	0	0	0	0	1	0	
SET_CONFIGURATION	0x00	9	n	0	0	0	0	0	n = 0,1
GET_INTERFACE	0x81	10	0	0	0-1	0	1	0	
CDC_ACM_IF SET_LINE_CODING	0x21	32	0	0	0	0	7 (Note 1)	0	Set the UART baud rate, parity, stop bits, etc.
CDC_ACM_IF GET_LINE_CODING	0xA1	33	0	0	0	0	7	0	Get the UART baud rate, parity, stop bits, etc.
CDC_ACM_IF SET_-CON-TROL_LINE_STATE	0x21	34	val (Note 2)	0	0	0	0	0	Set UART control lines
CDC_ACM_IF SEND_BREAK	0x21	35	val LSB	val MSB	0	0	0	0	Send a break for the specified duration

TABLE 2: SUPPORTED USB CONTROL COMMANDS

NAME	REQUEST TYPE	REQUEST	VALUE		INDEX		LENGTH		DESCRIPTION
XR_SET_REG	0x40	0	val LSB	val MSB	register addr. LSB	register addr. MSB	0	0	Exar custom command: set one 12-bit register val: 12-bit register value register address: see Table 5
XR_GET_REG	0xC0	1	0	0	register addr. LSB	register addr. MSB	2	0	Exar custom register: get one 12-bit register register address: see Table 5

NOTE: *1) Line coding length field are defined in* **Table 3**

NOTE: *2) Control Signal Bitmap values for SetControlLineState are defined in* **Table 4**

TABLE 3: SET_LINE_CODING

OFFSET	FIELD	SIZE	VALUE	DESCRIPTION
0	dwDTERate	4	Number	Data terminal rate, in bits per second
4	bCharFormat	1	Number	Stop bits: 0 = 1 Stop bit 2 = 1 Stop bits (1.5 stop bits not supported in B1411)
5	bParityType	1	Number	Parity: 0 = None 1 = Odd 2 = Even 3 = Mark 4 = Space
6	bDataBits	1	Number	Data bits (7, 8 or 9)

TABLE 4: SET_CONTROL_LINE_STATE

BIT POSITION	DESCRIPTION
D15..D2	Reserved (Reset to zero)
D1	Carrier control for half duplex modems. This signal corresponds to RS-232 signal RTS. 0 = Deactivate carrier (Clear RTS) 1 = Activate carrier (Set RTS) The device ignores the value of this bit when operating in full duplex mode
D0	Indicates to DCE if DTE is present or not. This signal corresponds to RS-232 signal DTR. 0 = Not present (Clear DTR) 1 = Present (Set DTR)

3.0 REGISTER SET DESCRIPTION

The internal register set of the B1411 controls the UART channel functionality, basic functionality of the FIFOs, OTP controls, as well as registers associated with the processing of driver commands. These registers are accessible via the USB interface using the XR_SET_REG and XR_GET_REG USB commands. 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. Several exceptions are the GPIO_SET and GPIO_CLEAR registers as well as the TX_BREAK and ERROR_STATUS registers. The UART does not need to be disabled when writing these four registers.

3.1 B1411 Register Map

TABLE 5: B1411 REGISTERS

ADDRESS	REGISTER NAME	BIT-11	BIT-10	BIT-9	BIT-8	BIT-7	BIT-6	BIT-5	BIT-4	BIT-3	BIT-2	BIT-1	BIT-0
0X20D	CUSTOM_DRIVER	0	0	0	0	0	0	0	0	0	0	0	ACTIVE
0x216	CDC_ACM_-FLOW_CONTROL	0	0	0	0	0	0	0	0	Half-Duplex	Flow Control Mode Select		
0x217	CDC_ACM_GPIO_-MODE	0	0	0	0	0	0	0	0	XCVR Enable Polarity	Mode Select		
0x218	CDC_ACM_GPIO_-DIRECTION	0	0	0	0	0	0	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
0x219	CDC_ACM_GPI-O_INT_MASK	0	0	0	0	0	0	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
0XC00	UART_ENABLE	0	0	0	0	0	0	0	0	0	0	RX	TX
0xC06	FLOW_CONTROL	0	0	0	0	0	0	0	0	Half-Duplex	Flow Control Mode Select		
0xC07	XON_CHAR	0	0	0	0	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0xC08	XOFF_CHAR	0	0	0	0	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0xC09	ERROR_STATUS	0	0	0	0	Break Status	Over-run Error	Parity Error	Fram-ing Error	Break Error	0	0	0
0xC0A	TX_BREAK	Bit-11	Bit-10	Bit-9	Bit-8	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0xC0B	RS485_DELAY	0	0	0	0	0	0	0	0	Delay			
0xC0C	GPIO_MODE	0	0	0	0	0	0	0	0	RS485 Polarity	Mode Select		
0xC0D	GPIO_DIRECTION	0	0	0	0	0	0	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
0xC0E	GPIO_SET	0	0	0	0	0	0	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
0xC0F	GPIO_CLEAR	0	0	0	0	0	0	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
0xC10	GPIO_STATUS	0	0	0	0	0	0	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0

TABLE 5: B1411 REGISTERS

ADDRESS	REGISTER NAME	BIT-11	BIT-10	BIT-9	BIT-8	BIT-7	BIT-6	BIT-5	BIT-4	BIT-3	BIT-2	BIT-1	BIT-0
0xC11	GPIO_INT_MASK	0	0	0	0	0	0	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
0xC12	CUSTOMIZED_INT	0	0	0	0	0	0	0	0	0	0	0	EN
0xC14	PIN_PULLUP_EN	0	0	0	0	TX	RX	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
0xC15	PIN_PULL-DOWN_EN	0	0	0	0	TX	RX	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1	GPIO 0
0xC16	LOOPBACK	0	0	0	0	0	0	0	0	0	DTR_ DSR	RTS_ CTS	TX_ RX
0XC80	TX_FIFO_RESET	0	0	0	0	0	0	0	0	0	0	0	TX
0xC81	TX_FIFO_COUNT	0	0	0	0	COUNT							
0XCC0	RX_FIFO_RESET	0	0	0	0	0	0	0	0	0	0	0	RX
0xCC1	RX_FIFO_COUNT	0	0	0	COUNT								
0xCC2	RX_FI-FO_LOW_LA-TENCY	0	0	0	0	0	0	0	0	0	0	0	EN
0xD02	WIDE_MODE	0	0	0	0	0	0	0	0	0	0	0	EN

3.1.1 B1411 Register Descriptions

Note that all register reset default values are ’0’ unless otherwise specified.

3.1.1.1 CUSTOM_DRIVER (Write Only)

CUSTOM_DRIVER[0]: Active

This register holds the flag to determine which device driver is used (custom or CDC driver). For proper operation, a custom driver must set the ACTIVE bit prior to sending any of the 4 CDC_ACM commands that the B1411 supports.

- Logic 0 = Informs the B1411 that the standard CDC_ACM driver is being used. Values from the CDC_ACM_xxx_xxxx registers will be loaded into their non-CDC_ACM equivalents.
- Logic 1 = Informs the B1411 that a custom driver is being used. Values from CDC_ACM_xxx_xxxx registers are not used.

CUSTOM_DRIVER[11:1]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.2 CDC_ACM_FLOW_CONTROL Register Description (Read / Write)

The contents of this register, if programmed, are used to overwrite the FLOW_CONTROL register at address 0xC06 when a CDC command is sent from a standard CDC-ACM driver to the B1411 device. Note that this register can only be programmed from the OTP. Since a standard CDC_ACM driver is unaware of UART registers in the B1411, this register may be utilized to program UART settings from power-up. When a custom driver is used, the custom driver should program these settings directly into the FLOW_CONTROL register.

Bit fields in this register are the same as those in the FLOW_CONTROL register. Refer to **“Section 3.1.1.7, FLOW_CONTROL Register Description (Read / Write)” on page 15**.

3.1.1.3 CDC_ACM_GPIO_MODE Register Description (Read / Write)

The contents of this register, if programmed, are used to overwrite the GPIO_MODE register at address 0xC0C when a CDC command is sent from a standard CDC-ACM driver to the B1411 device. Note that this register can only be programmed from the OTP. Since a standard CDC_ACM driver is unaware of UART registers in the B1411, this register may be utilized to program UART settings from power-up. When a custom driver is used, the custom driver should program these settings directly into the GPIO_MODE register.

Bit fields in this register are the same as those in the GPIO_MODE register. Refer to **"Section 3.1.1.13, GPIO_MODE Register Description (Read / Write)" on page 18**.

3.1.1.4 CDC_ACM_GPIO_DIRECTION Register Description (Read / Write)

The contents of this register, if programmed, are used to overwrite the GPIO_DIRECTION register at address 0xC0D when a CDC command is sent from a standard CDC-ACM driver to the B1411 device. Note that this register can only be programmed from the OTP. Since a standard CDC_ACM driver is unaware of UART registers in the B1411, this register may be utilized to program UART settings from power-up. When a custom driver is used, the custom driver should program these settings directly into the GPIO_DIRECTION register.

Bit fields in this register are the same as those in the GPIO_DIRECTION register. Refer to **"Section 3.1.1.14, GPIO_DIRECTION Register Description (Read / Write)" on page 18**.

3.1.1.5 CDC_ACM_GPIO_INT_MASK Register Description (Read / Write)

The contents of this register, if programmed, are used to overwrite the GPIO_INT_MASK register at address 0xC11 when a CDC command is sent from a standard CDC-ACM driver to the B1411 device. Note that this register can only be programmed from the OTP. Since a standard CDC_ACM driver is unaware of UART registers in the B1411, this register may be utilized to program UART settings from power-up. When a custom driver is used, the custom driver should program these settings directly into the GPIO_INT_MASK register.

Bit fields in this register are the same as those in the GPIO_INT_MASK register. Refer to **"Section 3.1.1.18, GPIO_INT_MASK Register Description (Read / Write)" on page 19**.

3.1.1.6 UART_ENABLE Register Description (Read / Write)

Ensure that both UART Tx and UART Rx are disabled before writing to any other UART registers except for the GPIO_SET, GPIO_CLEAR and Tx Break registers.

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[11:2]: Reserved

These bits are reserved and should remain '0'.

3.1.1.7 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 6: FLOW CONTROL MODE SELECTION

MODE	BIT-2	BIT-1	BIT-0	MODE DESCRIPTION
0	0	0	0	No flow control, no address matching.
1	0	0	1	HW flow control enabled. Auto RTS/CTS or DTR/DSR must be selected by GPIO_MODE.
2	0	1	0	SW flow control enabled
3	0	1	1	Multidrop mode - RX only after address match, TX independent. (Typically used with GPIO_MODE 3)
4	1	0	0	Multidrop mode - RX / TX only after address match. (Typically used with GPIO_MODE 4)

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[11:4]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.8 XON_CHAR Register Description (Read / Write - Default 0x17)

The XON_CHAR stores the XON character that is 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 1.3.7, Multidrop mode with address matching” on page 9**.

XON_CHAR[11:8]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.9 XOFF_CHAR Register Description (Read / Write - Default 0x19)

The XOFF_CHAR stores the XOFF character that is used in the Automatic Software Flow control.

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 1.3.7, Multidrop mode with address matching” on page 9**.

XOFF_CHAR[11:8]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.10 ERROR_STATUS Register Description - Read-clear

This register reports any historical errors that have occurred on the line such as break, framing, parity and overrun. Note that these errors cannot be directly associated with any bytes within the Rx FIFO. For diagnostic purposes, the WIDE_MODE can be enabled. In this mode, errors are real time, i.e. are directly associated with the current byte.

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.

ERROR_STATUS[11:8]: Reserved

- These bits are reserved and should remain ’0’.

3.1.1.11 TX_BREAK Register Description (Read / Write)

Writing a value between 1 and 0xFFE to this register causes a break condition to be generated continuously until the register is cleared. The register decrements at 1 ms intervals until the count is zero. If another non-zero value, other than 0xFFF is written to the TX_BREAK register before the counter decrements to zero, the decrement continues from the newly written value. A value of 0xFFF will cause the break condition to be generated until a different value is written to the register.

If data is being shifted out of the TX pin, the data will be completely shifted out before the break condition is generated.

Note that the break condition may be delayed by up to 1 ms following the write of the TX_BREAK register. Additionally, the break condition may persist for up to 2 bit times after the counter has decremented to zero.

3.1.1.12 RS485_DELAY Register Description (Read / Write)

RS485_DELAY[3:0]: Turn-around delay

This is the number of bit times the B1411 waits before de-asserting the GPIO5/RTS#/RS485 pin when it is configured for automatic RS-485 half-duplex control.

RS485_DELAY[11:4]: Reserved

These bits are reserved and should be ’0’.

3.1.1.13 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 1.3, UART” on page 5**.

TABLE 7: GPIO MODES

BITS [2:0]	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	MODE DESCRIPTION
000	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	GPIO Mode, All GPIO pins available as GPIO
001	GPIO0	GPIO1	GPIO2	GPIO3	CTS#	RTS#	GPIO4 and GPIO5 used for Auto RTS/CTS HW Flow Control
010	GPIO0	GPIO1	DSR#	DTR#	GPIO4	GPIO5	GPIO2 and GPIO3 used for Auto DTR/DSR HW Flow Control
011	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	RS485	GPIO5 used for auto RS-485 half-duplex control
100	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	RS485	GPIO5 used for auto RS-485 half-duplex control after address match (See FLOW_CONTROL mode 4).

GPIO_MODE[3]: RS485 Polarity

- Logic 0 = GPIO5/RTS#/RS485 Low for TX
- Logic 1 = GPIO5/RTS#/RS485 High for TX

GPIO_MODE[11:4]: 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.

3.1.1.14 GPIO_DIRECTION Register Description (Read / Write)

This register controls the direction of pins configured as GPIO. (Pins configured for UART functions via the GPIO_MODE register, e.g. RTS# are not controlled or reported in the GPIO_DIRECTION register.)

GPIO_DIRECTION[5:0]: GPIOx Direction

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

GPIO_DIRECTION[11:6]: Reserved

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

3.1.1.15 GPIO_SET Register Description (Read / Write)

Writing a ’1’ in this register sets the corresponding GPIO output high. Writing a ’0’ in this register sets the corresponding GPIO output low. For GPIO pins configured as an input via the GPIO_DIRECTION register this register has no effect. Bits 11-6 are unused and should be ’0’.

3.1.1.16 GPIO_CLEAR Register Description (Read / Write)

Writing a ’1’ in this register clears the corresponding GPIO output low. Writing a ’0’ to a bit has no effect. Bits 11-6 are unused and should be ’0’.

3.1.1.17 GPIO_STATUS Register Description (Read Only)

This register reports the current state of each of the GPIO pins.

3.1.1.18 GPIO_INT_MASK Register Description (Read / Write)

Dictates whether a change in GPIO pin state causes the device to generate a USB interrupt packet. In either case, the GPIO status register will still report the pin's state when read, and if an interrupt packet is formed due to other interrupt trigger, the interrupt packet will contain the current state of the pin.

GPIO_INT_MASK[5:0]: GPI0[5:0]

- Logic 0 = A change in the pin's state causes the device to generate an interrupt packet.
- Logic 1 = A change in the pin's state does not cause the device to generate an interrupt packet.

GPIO_INT_MASK[11:6]: Reserved

- These bits are reserved and should remain ’0’.

3.1.1.19 CUSTOMIZED_INT Register Description (Read / Write)

Enables the customized interrupt packet format to report all GPIO status in the interrupt packet.

CUSTOMIZED_INT[0]: Enable

- Logic 0 = Use standard interrupt packet. See Table 9
- Logic 1 = Use customized interrupt packet. See Table 10

CUSTOMIZED_INT[11:1]: Reserved

- These bits are reserved and should remain ’0’.

TABLE 8: INTERRUPT PACKET FORMAT

OFFSET	FIELD	SIZE (BYTES)	VALUE	DESCRIPTION
0	bmRequestType	1	8’b10100001	D7 = Device-to-host direction D6:5 = Class Type D4-0: = Interface Recipient
1	bNotification	1	8’h20	Defined encoding for SERIAL_STATE
2	wValue	2	16’h0000	
4	wIndex	2	16’h0000	D15-8 = Reserved (0) D7-0 = Interface number, 8’h00 for the CDC Command Interface
6	wLength	2	16’h0002	2 bytes of transferred data
8	Data	2	Standard int_status (See Table 9 or Table 10)	D15-7 = Reserved (0) D6 = bOverRun D5 = bParity D4 = bFraming D3 = bRingSignal (RI) D2 = bBreak D1 = bTxCarrier (DSR) D0 = bRxCarrier (CD)

TABLE 9: DATA FIELD OF STANDARD INTERRUPT PACKET

BITS	FIELD	DESCRIPTION
D15..D7		Reserved (future use)
D6	bOverRun	Received data has been discarded due to overrun in the device.
D5	bParity	A parity error has occured.
D4	bFraming	A framing error has occured.
D3	bRingSignal	State of ring signal detection of the device.
D2	bBreak	State of break detection mechanism of the device.
D1	bTxCarrier	State of transmission carrier. This signal corresponds to V.24 signal 106 and RS-232 signal DSR.
D0	bRxCarrier	State of receiver carrier detection mechanism of device. This signal corresponds to V.24 signal 109 and RS-232 signal DCD.

If the Exar vendor specific packet mapping is enabled then the data field also includes interrupt status for all of the UART / GPIO pins as follows:

TABLE 10: DATA FIELD OF CUSTOMIZED INTERRUPT PACKET - EXAR VENDOR SPECIFIC

BIT(S)	FIELD	DESCRIPTION
15	D15	Reserved (0)
14	D14	bGPIO5 (RTS)
13	D13	bGPIO4 (CTS)
12	D12	bGPIO3 (DTR)
11	D11	bGPIO0 (RI)
10	D10	Reserved (0)
9	D9	bGPIO2 (DSR)
8	D8	bGPIO1 (CD)
7	D7	Reserved (0)
6	D6	bOverRun
5	D5	bParity
4	D4	bFraming
3	D3	bRingSignal (RI)
2	D2	bBreak
1	D1	bTxCarrier (DSR)
0	D0	bRxCarrier (CD)

3.1.1.20 PIN_PULLUP_EN Register Description (Read / Write)

PIN_PULLUP_EN[5:0]: GPI0[5:0]

Enables internal pullup feature on the selected GPIO pins

- Logic 0 = Disable pullup on the corresponding pin.
- Logic 1 = Enable pullup on the corresponding pin - Caution: Do not enable pulldown simultaneously

PIN_PULLUP_EN[6]: UART Rx

Enables internal pullup feature on the UART Rx pin

- Logic 0 = Disable pullup on the corresponding pin.
- Logic 1 = Enable pullup on the corresponding pin - Caution: Do not enable pulldown simultaneously

PIN_PULLUP_EN[7]: UART Tx

Enables internal pullup feature on the UART Tx pin

- Logic 0 = Disable pullup on the corresponding pin.
- Logic 1 = Enable pullup on the corresponding pin - Caution: Do not enable pulldown simultaneously

PIN_PULLUP_EN[11:8]: Reserved

- These bits are reserved and should remain ’0’.

3.1.1.21 PIN_PULLDOWN_EN Register Description (Read / Write)

PIN_PULLDOWN_EN[5:0]: GPI0[5:0]

Enables internal pulldown feature on the selected GPIO pins

- Logic 0 = Disable pulldown on the corresponding pin.
- Logic 1 = Enable pulldown on the corresponding pin - Caution: Do not enable pullup simultaneously

PIN_PULLDOWN_EN[6]: UART Rx

Enables internal pulldown feature on the UART Rx pin

- Logic 0 = Disable pulldown on the corresponding pin.
- Logic 1 = Enable pulldown on the corresponding pin - Caution: Do not enable pullup simultaneously

PIN_PULLDOWN_EN[7]: UART Tx

Enables internal pulldown feature on the UART Tx pin

- Logic 0 = Disable pulldown on the corresponding pin.
- Logic 1 = Enable pulldown on the corresponding pin - Caution: Do not enable pullup simultaneously

PIN_PULLDOWN_EN[11:8]: Reserved

- These bits are reserved and should remain ’0’.

3.1.1.22 LOOPBACK Register Description (Read / Write)

LOOPBACK[0]: TX_RX

When this bit is set all transmitted UART data is looped back to the UART receiver. Note that when the internal loopback is enabled, the Tx data will be disabled and Rx data will be ignored.

- Logic 0 = Disable loopback.
- Logic 1 = Enable loopback.

LOOPBACK[1]: RTS_CTS

When this bit is set RTS is looped back to CTS.

- Logic 0 = Disable loopback.
- Logic 1 = Enable loopback.

LOOPBACK[2]: DTR_DSR

When this bit is set DTR is looped back to DSR.

- Logic 0 = Disable loopback.
- Logic 1 = Enable loopback.

LOOPBACK[11:3]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.23 TX_FIFO_RESET (Write Only)

TX_FIFO_RESET[0]: Reset

- Write a ’1’ to reset the Tx FIFO, self-clearing.

TX_FIFO_RESET[11:1]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.24 TX_FIFO_COUNT (Read Only)

TX_FIFO_COUNT[7:0]: Character Count

- Reports the number of characters currently in the Tx FIFO.

TX_FIFO_COUNT[11:8]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.25 RX_FIFO_RESET (Write Only)

RX_FIFO_RESET[0]: Reset

- Write a ’1’ to reset the Rx FIFO, self-clearing.

RX_FIFO_RESET[11:1]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.26 RX_FIFO_COUNT (Read Only)

RX_FIFO_COUNT[8:0]: Character Count

- Reports the number of characters currently in the Rx FIFO.

RX_FIFO_RESET[11:9]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.27 RX_FIFO_LOW_LATENCY (Read / Write)

RX_FIFO_LOW_LATENCY[0]: Low Latency Enable

This register is automatically set to logic ’1’ for baud rates below 40961 bps.

- Logic 0 = Receive data is not from Rx FIFO until bMaxPacketSize (normally 64 bytes) or timeout (3 characters) has been reached. (Note: When the CDC-ACM driver is used, the bMaxPacketSize becomes 63 bytes.)
- Logic 1 = Receive data is forwarded from Rx FIFO immediately upon receipt.

RX_FIFO_LOW_LATENCY[11:1]: Reserved

These bits are reserved and should remain ’0’.

3.1.1.28 WIDE_MODE (Read / Write)

WIDE_MODE[0]: EN

- Logic 0 = Normal (7, 8 or 9 bit data) mode
- Logic 1 = Wide mode - See **“Section 1.3.1.1, Wide mode Transmit” on page 6**, **“Section 1.3.2.3, Wide mode operation with 7 or 8-bit data” on page 6** and **“Section 1.3.2.4, Wide mode operation with 9-bit data” on page 6**.

WIDE_MODE[11:1]: Reserved

These bits are reserved and should remain ’0’.

3.2 OTP Memory

The OTP on-chip memory contents are accessible via the USB interface. For details on programming the OTP contact uarttechsupport@exar.com. Note that certain memory locations are pre-programmed at the factory. Programming any of these locations or locations not documented in the data sheet may cause permanent functional damage to the B1411 device.

TABLE 11: OTP MEMORY

ADDRESS	REGISTER NAME	BIT-7	BIT-6	BIT-5	BIT-4	BIT-3	BIT-2	BIT-1	BIT-0
0x00	OTP_CONFIG0	FACTORY PROGRAMMED - DO NOT OVERWRITE						LOWPOWER_POL	RESERVED
0x01	OTP_CONFIG1	FACTORY PROGRAMMED - DO NOT OVERWRITE							
0x02	OTP_CONFIG2	FACTORY PROGRAMMED - DO NOT OVERWRITE							
0x03	OTP_CONFIG3	0	0	0	0	ENA_VBUS_SENSE	CORE_CLOCK_DIV		
0x04	OTP_VALID	0	0	0	0	0	LINE_CODING	STRINGS	USB
0x05 - 0x0B	LINE_CODING_0 - LINE_CODING_6	VALUE							
0x0C	USB_VENDOR_ID_LSB	VALUE							
0xD	USB_VENDOR_ID_MSB	VALUE							
0xE	USB_PRODUCT_ID_LSB	VALUE							
0xF	USB_PRODUCT_ID_MSB	VALUE							
0x10	USB_ATTRIBUTES	0	SELF-_POWERED	REMOTE_WAKEUP	RESERVED				
0x11	USB_MAXPOWER	VALUE							
0x12 - 0X21	MANUFACTURER_STRING_0 - MANUFACTURER_STRING_15	VALUE							
0X22 - 0X31	PRODUCT_STRING_0 - PRODUCT_STRING_15	VALUE							
0x32 - 0X41	SERIAL_NUMBER_STRING_0 - SERIAL_NUMBER_STRING_15	VALUE							
0x42 - 0x1FF	ADDRESS_VALUE	VALUE							

3.2.1 OTP Memory Descriptions

Some OTP memory locations are pre-programmed at the factory before shipments to customers. Programming any memory location other than those specified below may result in permanent damage to the device.

3.2.1.1 OTP Config0 (Read / Write OTP)

OTP_CONFIG0[0]: Reserved

- Factory programmed - overwriting this bit may cause functional damage to the B1411 device

OTP_CONFIG0[1]: Lowpower_Pol

- Sets the polarity of the LOWPOWER output pin
  - •Logic 0 = LOWPOWER output pin will be active low
  - •Logic 1 = LOWPOWER output pin will be active high

OTP_CONFIG0[7:2]: Reserved

- Factory programmed - overwriting these bits may cause functional damage to the B1411 device

3.2.1.2 OTP Config1 (Read / Write OTP)

OTP_CONFIG1[7:0]: Reserved

- Factory programmed - overwriting these bits may cause functional damage to the B1411 device

3.2.1.3 OTP Config2 (Read / Write OTP)

OTP_CONFIG2[7:0]: Reserved

- Factory programmed - overwriting these bits may cause functional damage to the B1411 device

3.2.1.4 OTP Config3 (Read / Write OTP)

OTP_CONFIG3[2:0]: Core_Clock_Select

The B1411 core can run at a fraction of the 48 MHz bus clock to minimize power consumption in the core. Refer to **Table 12** for core clock divider settings. Note that the selected core clock rate must be a minimum of 4x the maximum baud rate setting desired in a customer application. For example, if a core clock of 6 MHz is selected, the maximum baud rate of the B1411 is 1.5 Mbps.

TABLE 12: CORE CLOCK DIVIDER

VALUE	NAME	DESCRIPTION
3'b000	DIV_BY_1	Core Clock = CLOCK / 1 (48 MHz)
3'b001	DIV_BY_2	Core Clock = CLOCK / 2 (24 MHz)
3'b010	DIV_BY_4	Core Clock = CLOCK / 4 (12 MHz)
3'b011	DIV_BY_8	Core Clock = CLOCK / 8 (6 MHz)
3'b100 - 3'b111	Not Used	Reserved - Using these settings may cause functional damage to the B1411 device

OTP_CONFIG3[3]: Ena_VBUS_Sense

- Controls whether VBUS is sensed.
  - •Logic 0 = VBUS sense is not enabled (typically used in bus-powered mode)
  - •Logic 1 = VBUS sense is enabled (typically used in self-powered mode)

OTP_CONFIG3[7:4]: Reserved

These bits are reserved and should remain ’0’.OTP Valid (Read / Write OTP)

3.2.1.5 OTP _Valid (Read / Write OTP)

This register holds the VALID flag for the OTP override values. These include the USB device and configuration descriptor overrides and default line coding overrides.

OTP_VALID[0]: USB

- Set this bit to indicate that the USB device and configuration descriptors have selected fields overwritten by OTP data.

OTP_VALID[1]: Strings

- Set this bit to indicate that the contents of the USB string descriptors are overwritten by OTP data.

OTP_VALID[2]: Line_Coding

- Set this bit to indicate that the power up line coding (baud rate, stop bits, parity) are overwritten by OTP data.

OTP_VALID[7:3]: Reserved

- These bits are reserved and should remain ’0’

3.2.1.6 Line Coding0-6 (Read / Write OTP)

LINE_CODING0-6[7:0]: Value

The contents of this field have the same format as that used in the CDC-ACM SET_LINE_CODING request. It allows the default baud rate and character format to be overridden.

3.2.1.7 USB_VENDOR_ID_LSB (Read / Write OTP)

USB_VENDOR_ID_LSB[7:0]: Value

Bits [7:0] of the USB vendor ID reported with the descriptors. If OTP_VALID.USB is set to 0, then the Exar vendor ID is reported instead.

3.2.1.8 USB_VENDOR_ID_MSB (Read / Write OTP)

USB_VENDOR_ID_MSB[7:0]: Value

Bits [15:8] of the USB vendor ID reported with the descriptors. If OTP_VALID.USB is set to 0, then the Exar vendor ID is reported instead.

3.2.1.9 USB_PRODUCT_ID_LSB (Read / Write OTP)

USB_PRODUCT_ID_LSB[7:0]: Value

Bits [7:0] of the USB product ID reported with the descriptors. If OTP_VALID.USB is set to 0, then the Exar product ID is reported instead.

3.2.1.10 USB_PRODUCT_ID_MSB (Read / Write OTP)

USB_PRODUCT_ID_MSB[7:0]: Value

Bits [15:8] of the USB product ID reported with the descriptors. If OTP_VALID.USB is set to 0, then the Exar product ID is reported instead.

3.2.1.11 USB_ATTRIBUTES (Read / Write OTP)

USB_ATTRIBUTES[4:0]: Reserved

- These bits are reserved and should remain ’0’