



# Digital UART

## Product Specification

PS038902-1116



**Warning:** DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

---

### **LIFE SUPPORT POLICY**

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

### **As used herein**

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

### **Document Disclaimer**

©2016 Zilog, Inc All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

ZMOTION and Z8 Encore! XP are trademarks or registered trademarks of Zilog, Inc. (An IXYS Company). All other product or service names are the property of their respective owners.

# Revision History

Each instance in the revision history table reflects a change to this document from its previous revision. For more details, refer to the corresponding pages or appropriate links provided in the table below.

| Date     | Revision Level | Description  | Page   |
|----------|----------------|--|--|
| Nov 2016 | 02             | Corrected power supply voltage; updated Functional Description; updated Table 2; added a note for the Read System Status Register. | <a href="#">2</a> ; <a href="#">10</a> ; <a href="#">15</a> ; <a href="#">39</a> |
| Feb 2016 | 01             | Original issue.  | All  |

# Table of Contents

|  |     |
|--|-----|
| Revision History .....                       | iii |
| Table of Contents .....                      | iv  |
| List of Figures .....                        | vi  |
| List of Tables .....                         | vii |
| Overview .....                               | 1   |
| Features .....                               | 2   |
| Applications .....                           | 4   |
| Block Diagram .....                          | 5   |
| Pin Description .....                        | 6   |
| Pin Configurations .....                     | 6   |
| Pin Descriptions .....                       | 8   |
| Functional Description .....                 | 10  |
| I <sup>2</sup> C .....                       | 10  |
| Stacked Write Commands .....                 | 10  |
| I <sup>2</sup> C Addressing .....            | 10  |
| I <sup>2</sup> C Protocol Overview .....     | 10  |
| UART .....                                   | 11  |
| Baud Rate .....                              | 11  |
| Flow Control .....                           | 12  |
| Watermark .....                              | 12  |
| Interrupt .....                              | 13  |
| EEPROM .....                                 | 13  |
| I <sup>2</sup> C EEPROM Operations .....     | 13  |
| GPIO .....                                   | 14  |
| Commands .....                               | 15  |
| Command List .....                           | 15  |
| EEPROM Command Detail .....                  | 17  |
| Write EEPROM at Current Location .....       | 17  |
| Read EEPROM at Current Location .....        | 17  |
| Write EEPROM Current Location Register ..... | 17  |
| Read EEPROM Current Location Register .....  | 18  |
| Erase EEPROM .....                           | 18  |

|  |    |
|--|----|
| GPIO Command Detail . . . . .                            | 19 |
| Setting GPIO Out Register . . . . .                      | 19 |
| Reading GPIO In Register . . . . .                       | 20 |
| Write GPIO Configuration . . . . .                       | 20 |
| Read GPIO Configuration . . . . .                        | 25 |
| Read GPIO Interrupt Status Register . . . . .            | 28 |
| UART Command Detail . . . . .                            | 28 |
| Read UART Status Register . . . . .                      | 28 |
| Enable Interrupts . . . . .                              | 29 |
| Interrupt Status Register . . . . .                      | 30 |
| Write Data to TX FIFO . . . . .                          | 30 |
| Read RX FIFO Data . . . . .                              | 30 |
| Write Baud Rate Register . . . . .                       | 31 |
| Read Actual Baud Rate Register . . . . .                 | 31 |
| Write Configuration . . . . .                            | 32 |
| Read Configuration . . . . .                             | 36 |
| Write Transmit Watermark Register . . . . .              | 37 |
| Read Transmit Watermark Register . . . . .               | 37 |
| Write Receive Watermark Register . . . . .               | 37 |
| Read Receive Watermark Register . . . . .                | 38 |
| Enable UART . . . . .                                    | 38 |
| Read Receive and Transmit FIFO Level Registers . . . . . | 38 |
| System Command Detail . . . . .                          | 39 |
| Read System Status Register . . . . .                    | 39 |
| Read Last Operation Result Register . . . . .            | 39 |
| Read System Version . . . . .                            | 39 |
| Read Interrupt Source Register . . . . .                 | 40 |
| Electrical Characteristics . . . . .                     | 41 |
| Absolute Maximum Ratings . . . . .                       | 41 |
| Recommended Operating Conditions . . . . .               | 42 |
| Power-on Reset Characteristics . . . . .                 | 42 |
| Internal RC Oscillator Characteristics . . . . .         | 43 |
| DC Characteristics . . . . .                             | 43 |
| Packaging . . . . .                                      | 44 |
| Ordering Information . . . . .                           | 48 |
| Customer Support . . . . .                               | 49 |



# List of Tables

Table 1. Pin Descriptions ..... 8  
Table 2. Command List ..... 15  
Table 3. Ordering Information ..... 48

# List of Figures

- Figure 1. Digital UART Block Diagram ..... 5
- Figure 2. ZDU0110RFX Pin Configuration ..... 6
- Figure 3. ZDU0110RHX Pin Configuration ..... 6
- Figure 4. ZDU0110RJX Pin Configuration ..... 7
- Figure 5. ZDU0110QUX Pin Configuration ..... 7
- Figure 6. 32-Pin QFN Package ..... 44
- Figure 7. 28-Pin TSSOP Package ..... 45
- Figure 8. 20-Pin TSSOP Package ..... 46
- Figure 9. 16-Pin TSSOP Package ..... 47

# Overview

Zilog's Digital Universal Asynchronous Receiver/Transmitter (UART) is a single-chip CMOS communications device which provides full duplex asynchronous communications with a 128 byte FIFO buffer, of which 64 bytes each are allocated to Receive and Transmit operations. This device also contains a 4 kbit EEPROM and General-Purpose Input and Output (GPIO) with programmable interrupt capability. Zilog's Digital UART has an internal 8 MHz oscillator, which eliminates the requirement for an external crystal oscillator.

This Digital UART is controlled by the simple I<sup>2</sup>C protocol (2-Wire Interface), allowing up to eight devices in the same I<sup>2</sup>C network. Although this device is fully programmable through the exposed commands, it is preconfigured to operate the UART at a 57,600 baud rate; therefore, configuration is not required to access the UART or the EEPROM.

Zilog's Digital UART provides separate programmable interrupts and interrupt lines for UART notification and GPIO notifications, so the controlling device does not have to poll the device for data.



# Features

Zilog's Digital UART offers the following features:

- Single power supply from 1.8V to 5.0V
- No external crystal needed
- Minimal pin count for space saving
- Interrupt lines for UARTs and GPIOs for notification
- Control via I<sup>2</sup>C standard protocol, up to 400K
  - Up to 8 different addresses available
  - Can stack up to 8 parts on same I2C network
  - Ability to send multiple write commands in a single transfer
- No configuration necessary
  - Default UART setting: 57600 baud, no flow control, no parity, 8 bits
  - EEPROM is accessed through standard I2C EEPROM communications
  - GPIO configured as inputs
- Up to two industry standard full duplex individual UART channels
  - 64 byte receiver FIFO for each UART
  - 64 byte transmitter FIFO for each UART
  - Flow control using CTS and RTS pins
  - In band flow control using XON/XOFF
  - FIFO watermark settings for configuring interrupt lines
  - Programmable data formats
    - 5 to 8 bits data
    - Odd, Even or no parity
    - 1 or 2 stop bits
    - Loopback for testing
  - Flexible baud rate selection, in steps of 100 bits per second, up to 250K
  - Programmable Interrupts
- 4KB built-in I<sup>2</sup>C EEPROM
  - Standard I<sup>2</sup>C EEPROM communications
  - 300K times endurance
  - 10 year retention
- General Purpose Input Output pins

- Up to 12 GPIO pins
- 2 pins can generate interrupt on change of pin value
- Each pin is fully configurable
  - Individual pin configuration
  - Direction (input/output)
  - Open drain output option
  - Weak pull up option
  - Debounce option
  - Quickly return to default settings

# *Applications*

The Digital UART is suitable for use in multiple applications; examples include:

- Point of Sale devices
- Communication bridges
- Process control
- Building/automation control
- Terminal servers
- Print systems

# Block Diagram

Figure 1 shows a block diagram of the Digital UART.

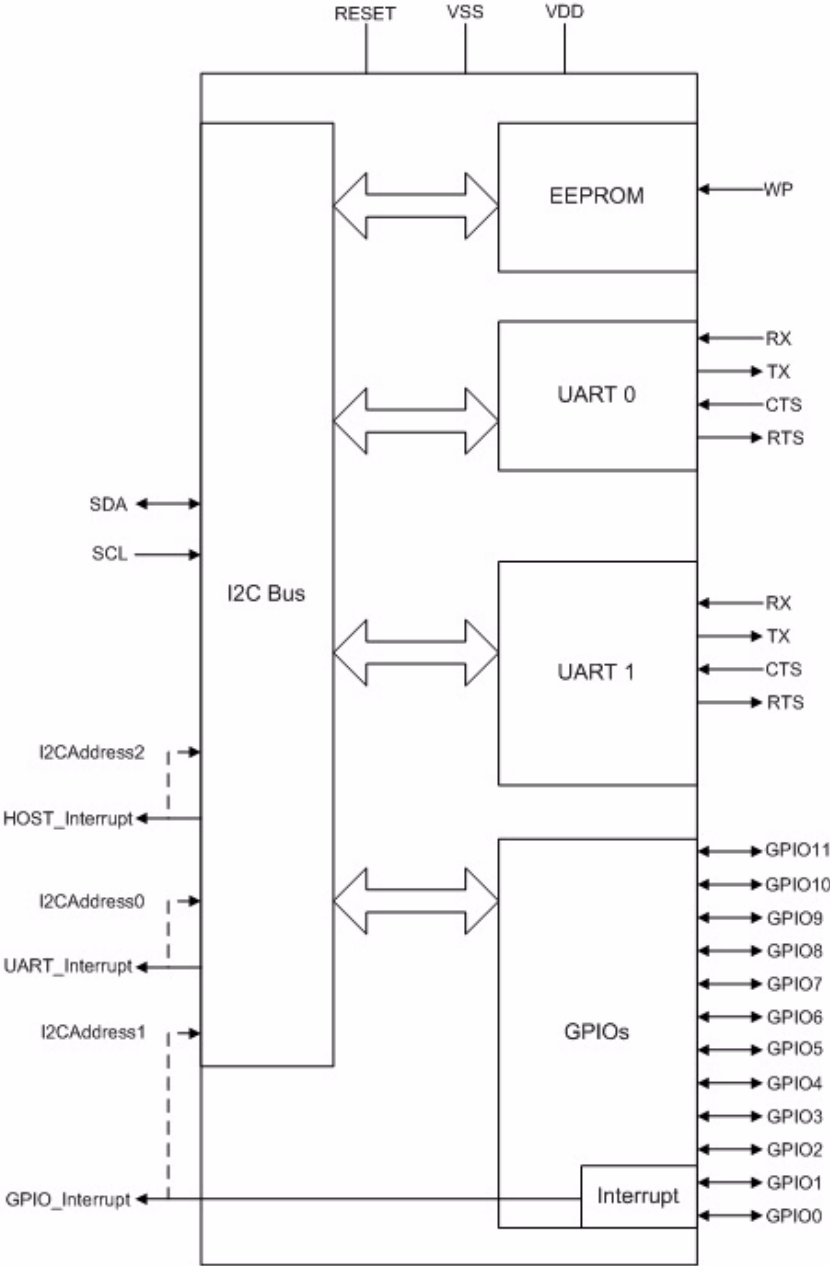


Figure 1. Digital UART Block Diagram

# Pin Description

The Digital UART is available in a variety of package styles and pin configurations. This chapter describes the available pin configurations for each of the package style. To learn more about the physical package specifications, see the [Ordering Information](#) chapter on page 48.

## Pin Configurations

Figures 2 through 5 show the pin configurations for each Digital UART package style.

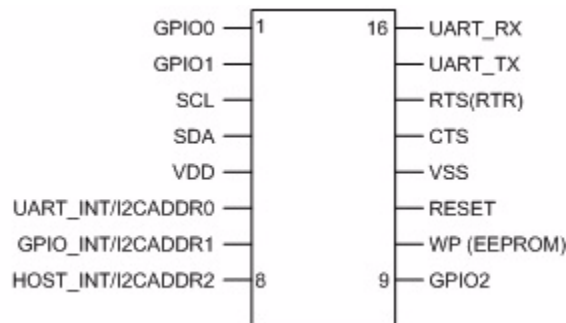


Figure 2. ZDU0110RFX Pin Configuration

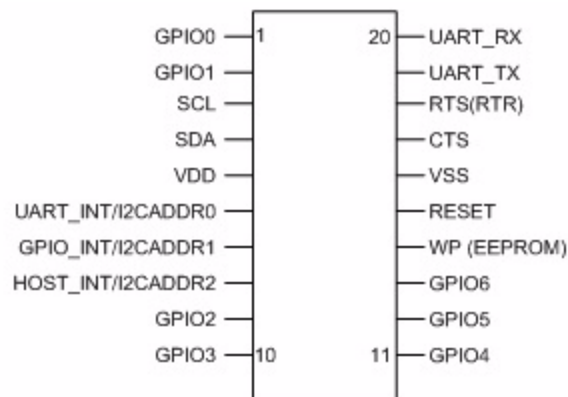


Figure 3. ZDU0110RHX Pin Configuration

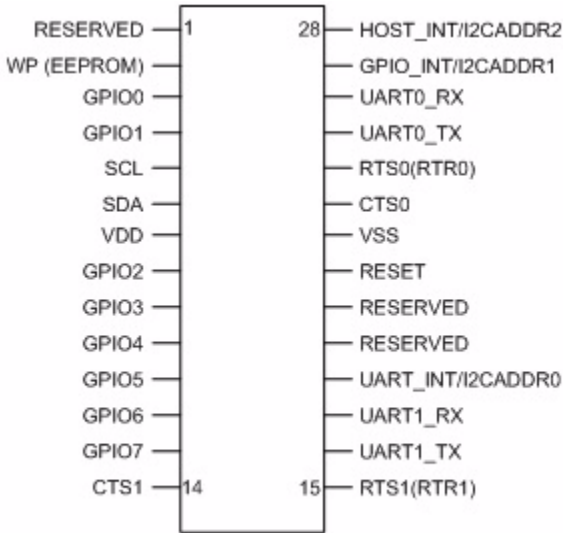


Figure 4. ZDU0110RJX Pin Configuration

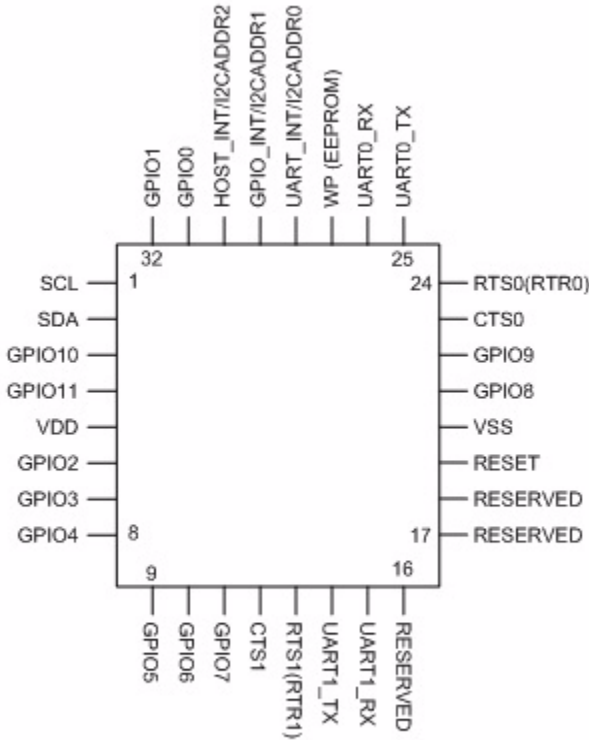


Figure 5. ZDU0110QUX Pin Configuration

## Pin Descriptions

[Table 1](#) lists the pins and their descriptions.

**Table 1. Pin Descriptions**

| Pin Number |    |    |    | Pin Name          | Description                           | Direction |
|------------|----|----|----|-------------------|---------------------------------------|-----------|
| 16         | 20 | 28 | 32 |                   |                                       |           |
| 3          | 3  | 5  | 1  | SCL               | I2C Serial Clock                      | IN        |
| 4          | 4  | 6  | 2  | SDA               | I2C Serial Data                       | IN/OUT    |
| 6          | 6  | 18 | 28 | UART_INT/I2CADDR0 | UART Interrupt Line <sup>1</sup>      | OUT       |
| 7          | 7  | 27 | 29 | GPIO_INT/I2CADDR1 | GPIO Interrupt Line <sup>1</sup>      | OUT       |
| 8          | 8  | 28 | 30 | HOST_INT/I2CADDR2 | HOST Interrupt Line <sup>1,2</sup>    | OUT       |
| 15         | 19 | 25 | 25 | UART0_TX          | UART 0 Transmit                       | OUT       |
| 16         | 20 | 26 | 26 | UART0_RX          | UART 0 Receive                        | IN        |
| 13         | 17 | 23 | 23 | CTS0              | UART 0 Clear to Send                  | IN        |
| 14         | 18 | 24 | 24 | RTS0(RTR0)        | UART 0 Request to Send                | OUT       |
| –          | –  | 16 | 14 | UART1_TX          | UART 1 Transmit                       | OUT       |
| –          | –  | 17 | 15 | UART1_RX          | UART 1 Receive                        | IN        |
| –          | –  | 14 | 12 | CTS1              | UART 1 Clear to Send                  | IN        |
| –          | –  | 15 | 13 | RTS1(RTR1)        | UART 1 Ready to Send                  | OUT       |
| 10         | 14 | 2  | 27 | WP (EEPROM)       | Write Protect for EEPROM <sup>3</sup> | IN        |
| 1          | 1  | 3  | 31 | GPIO0             | General purpose I/O with Int          | IN/OUT    |
| 2          | 2  | 4  | 32 | GPIO1             | General purpose I/O with Int          | IN/OUT    |
| 9          | 9  | 8  | 6  | GPIO2             | General purpose I/O                   | IN/OUT    |
| –          | 10 | 9  | 7  | GPIO3             | General purpose I/O                   | IN/OUT    |
| –          | 11 | 10 | 8  | GPIO4             | General purpose I/O                   | IN/OUT    |
| –          | 12 | 11 | 9  | GPIO5             | General purpose I/O                   | IN/OUT    |
| –          | 13 | 12 | 10 | GPIO6             | General purpose I/O                   | IN/OUT    |
| –          | –  | 13 | 11 | GPIO7             | General purpose I/O                   | IN/OUT    |
| –          | –  | –  | 21 | GPIO8             | General purpose I/O                   | IN/OUT    |
| –          | –  | –  | 22 | GPIO9             | General purpose I/O                   | IN/OUT    |
| –          | –  | –  | 3  | GPIO10            | General purpose I/O                   | IN/OUT    |

**Table 1. (Continued)Pin Descriptions**

| Pin Number |    |    |    | Pin Name | Description                          | Direction |
|------------|----|----|----|----------|--------------------------------------|-----------|
| 16         | 20 | 28 | 32 |          |                                      |           |
| –          | –  | –  | 4  | GPIO11   | General purpose I/O                  | IN/OUT    |
| 11         | 15 | 21 | 19 | RESET    | Active Low Reset                     | IN        |
| 5          | 5  | 7  | 5  | VDD      | Positive Supply                      | N/A       |
| 12         | 16 | 22 | 20 | VSS      | Ground                               | N/A       |
| –          | –  | *  | *  | RESERVED | Reserved for future use <sup>4</sup> | N/A       |

1 Interrupt lines are shared with the I2C Address selection. When system comes out of reset (either by bringing Reset pin low or operations.

2. Host Interrupt line is currently not used, Reserved for future use

3. Write Protect for EEPROM is to prevent writing to EEPROM. To Write to the EEPROM, this pin must be held low.

4. Reserved for future use pins are not used and should be pulled up to VDD



## Functional Description

Upon power-up, the Digital UART device reads the I<sup>2</sup>C addresses for the correct configuration and addressing. The system then asserts all interrupt pins, configures the I<sup>2</sup>C host interfaces, configures all the peripherals to the default configurations and then de-asserts all interrupts, notifying the host that the initialization is completed. While the interface is being configured, communications are not possible; however, after the host interface has been configured, the system will respond to a system status command while the rest of the system is being initialized. After all write transactions, the I<sup>2</sup>C will NAK any start packet until the write command has been processed, including the system status command.

### I<sup>2</sup>C

The Digital UART is an I<sup>2</sup>C slave device using a 7 bit address (Bit 0 specifies the Read or Write operation). The maximum bus speed that the Digital UART can support is 400 kHz. This device has up to eight possible addresses, allowing up to 8 devices on a single I<sup>2</sup>C bus. The I<sup>2</sup>C bus uses two bi-directional open-drain lines, pulled up to VDD with resistors. All I<sup>2</sup>C transactions must be separated by a wait period of at least 4 microseconds.

### Stacked Write Commands

The Digital UART supports stacked write requests for multiple commands at the same time (up to a 64-byte packet). A *stacked* packet allows the Host to use one transaction to write to multiple commands, such as when configuring UART and/or GPIOs. If there is an error in the packet, packet processing is stopped and the error condition is logged in the System Status Register until the next request is processed.

### I<sup>2</sup>C Addressing

The device responds to the following two addresses:

- 1010XXXb where XXX is the address configured using the I2CADDR pins  
To be used to access the EEPROM through de facto standard interface
- 1011XXXb where XXX is the address configured using the I2CADDR pins  
To be used to access the commands through standard I<sup>2</sup>C protocol.

### I<sup>2</sup>C Protocol Overview

The Master is responsible for generating the serial clock (SCL) and generates the start and stop conditions on the data line (SDA). The Master can only send data when the bus is not busy, as defined by both data and clock lines remaining high.

A transfer starts with a High to Low transition of the SDA line while the SCL is high. The state of the data line is valid data when the SCL line is high. The SDA can only be changed while the SCL is low. The receiving device can keep the SCL line low to suspend the transaction. After the receiving device releases the SCL line, the transaction can continue. Each byte must be acknowledged by the receiving device.

To acknowledge the byte, the receiving device pulls down the SDA during the acknowledge clock pulse so that while the clock pulse is in a low state, the SDA is in a stable low state when the clock is high. If the SDA is high during the acknowledge cycle, it is considered *Not Acknowledged*, signaling the end of the packet. The Master then follows with a stop condition, transitioning a Low to High on the SDA line while the clock is high.

The first byte following the Start condition from the master device is an address byte. The address byte consists of 7 bits plus a Read/Write (R/W) bit. If the address sent was acknowledged, then a slave device responded, so the transaction can continue. If an acknowledge was not received, the master device issues a stop condition and ends the transaction.

If the R/W bit was a Read request, then the master sends 8 pulses to receive the data. If the master has more to read during this transaction, the master acknowledges the byte and repeats the process. After the requested data is completed, the master sends a stop condition to end the transaction.

If the R/W bit was a Write request, the master continues to send the data until either all the data was sent or a byte was not acknowledged. After the write transaction is complete, if a read transaction is necessary the master device can send a Start condition (without the Stop condition) to change to a read request. When the transaction is complete, a Stop condition is sent to end the transaction.

## UART

The UART peripheral is a fully programmable, flexible, and preconfigured serial port. When powered up, the default setting is 57600 baud rate, 8 bits, no parity, no flow control. This setting allows the host to send and receive data as soon as the Digital UART is operational, with no configuration necessary. The UART peripheral can be configured for different baud rates in multiples of 100, up to 230K. The UART can also be configured for 5,6,7, or 8 bit data transfers; multiple flow control options; 1 or 2 stop bits; Odd, Even or no parity. The UART also has the ability to set different conditions to notify the host through the interrupt line.

### Baud Rate

The baud rate is configured in units of 100; however, due to the 8MHz clock, the actual rate may be different. Use the following equation to calculate the actual baud rate:

$$\frac{10000}{\text{Round}(10000 \div (\text{requestedbaud}))} \times 100$$

The baud rate in the register will be the integer of actual baud rate divided by 100.

**Example:**

|                      |   |
|----------------------|---|
| Requested Baud Rate: | 576 (57600 / 100)                       |
| Actual Rate:         | 10000/(Round(10000/576)) * 100 = 58,823 |
| Baud Rate Register   | 588                                     |

## Flow Control

Two types of flow control are available on the Digital UART. Flow control provides the signaling between the UART connections to indicate that it is safe to send data. Flow control is recommended for maximum reliability and the preferred method is hardware flow control. For flow control to work, both devices must be configured for the same type of flow control.

### Hardware Flow Control

Hardware flow control provides two extra pins to enable the sending/receiving of data from the UART TX/RX pins. The local Clear to Send (CTS) pin is connected to the remote Request to Send (RTS) pin. The Remote CTS is connected to the Local RTS. The CTS pin is an active low input pin, while the RTS pin is an output pin.

When the receiver is ready to receive data, the receiving UART brings the RTS pin (which is connected to the transmitter's CTS pin) low. The CTS pin going low signals the transmitter to send the data.

### Software Flow Control

Software Flow control is signaled through in-band communications. The receiver controls the transmission by sending the XON (0x11) and XOFF (0x13) characters to the transmitter. When the receiver is not ready to receive a byte, it sends an XOFF character to notify the transmitter not to send any more data until it receives an XON character. Because this is in-band, the transmitter may not see the XOFF character immediately (it must receive and process the character), and there may be some loss if the speed is too fast. The Digital UART sends an XOFF character every time it receives a byte and sends an XON character when it is ready to receive more data. This helps minimize the potential loss of in-band signaling.

## Watermark

The Digital UART contains a 64 byte buffer for Receive characters and a separate 64 byte buffer for Transmit characters. A configurable *watermark*, also known as a *trigger*, notifies the host about the position of the data at the set point, via the UART Interrupt. This

does not stop the buffer from filling beyond the watermark, it only notifies the host that the watermark level has been reached.

For example, if you only want to be notified after the receive buffer is half-filled, set the receive watermark to 32 and configure the interrupt to notify the host when the receive watermark is reached. The host will not be notified until the watermark interrupt is received, thereby freeing it up to perform other processing tasks. After the watermark interrupt is received, the host can retrieve all the bytes of received data at the same time. The receive buffer continues to fill beyond the 32 bytes and the UART interrupt remains asserted as long as there is 32 bytes or more in the buffer.

## Interrupt

The Digital UART provides a separate interrupt line (output) for all the UARTs on the device. There are 8 selectable interrupts that can assert the UART interrupt line. The host can configure which interrupts are of interest. When the condition is reached, the Digital UART asserts the interrupt line (by pulling the line low) to notify the host of the condition. The host then queries the interrupt statuses for the UARTs to identify the condition(s) reached and on which UART. All interrupt conditions are maintained, even if the host has not configured to be notified of the condition. Thus, the Digital UART features the ability to poll for any interrupt condition and only be notified of selected conditions.

## EEPROM

The 512 byte EEPROM is accessible via I<sup>2</sup>C communications. There are two ways to set/get the data:

- Use the commands to read, write, and erase the EEPROM, using the I<sup>2</sup>C address of 1011xxxxb (See the [Commands](#) chapter on page 15).
- Use the de facto standard of most I<sup>2</sup>C EEPROMs. This uses the address 1010xxxxb.

The EEPROM uses 32 byte pages with a total of 16 pages.

## I<sup>2</sup>C EEPROM Operations

### Write Operation

To write to the EEPROM, the Write Protect (WP) pin must be a digital low. The host can hold this high until ready to write, then pull low, write to the data, then pull high to continue when the writing is complete, without concern of writing EEPROM unintentionally.

Following the 7 bit address (1010xxxxb), the write bit (logic low) is sent from the master, thus completing the 8 bit packet. The slave device generates an acknowledge bit during the ninth clock cycle. The next two bytes sent by the master are the address to write the

data to. The first byte is the high byte of the word, followed by the low byte, then the data to write at that address (max of 32 bytes if the WP pin is low). When the data is complete, the master sends a stop condition, signaling the end of packet. After every byte, the slave device acknowledges the byte after it has been processed (not written). After the stop condition is received, the slave then starts the writing of the data to the EEPROM at the location specified (if the WP pin is low). The slave does not acknowledge any further requests until the write cycle is completed. The master can poll the device for completion by sending the address with the R/W bit set to a logical 0. If the cycle is in process, the slave will not acknowledge the request.

## Read Operation

### Read from Current Location

The master sends the start condition followed by the 7 bit address. The read bit (logic high) is sent (completing the 8 bit packet) and the slave device generates the acknowledge bit during the ninth cycle when it is ready to send. The slave then starts transmitting the bytes located at the current address pointer. After each byte is sent, the current address pointer is incremented. The address pointer rolls to 0 when incremented past max bytes. After each byte, the master sends an acknowledgement during the time it wants to continue receiving data. When the master has completed receiving the data it requires, no acknowledgement is sent; instead, the stop condition is sent, concluding the transfer.

### Random Read

Random read allows data to be read from anywhere by sending an address to set the current address pointer to the beginning of the data to be read. To set the current address pointer, the master issues a start condition, followed by the 7 bit address and a write bit (logic low). The slave device acknowledges the byte and the master then sends the word address (first byte, high byte; second byte, low byte) to set the current address pointer to. The slave device acknowledges both bytes. The master then issues another start condition following the last acknowledgment, which terminates the write operation. Then the master issues the address with the read bit set and continues with the read from current location.

## GPIO

The Digital UART provides up to 12 GPIO pins. Two of the GPIO pins (GPIO0, GPIO1) are able to cause the GPIO interrupt line to be asserted when the value on the input pin is toggled.

Each pin can be configured for input, output (push-pull or open drain), weak pull up resistor and de-bounce capability.

# Commands

The commands identify peripherals using the high 3 bits and the lower 5 bits as the command. The EEPROM/GPIO peripheral uses 000b, UART0 uses 001b, UART1 uses 010b and System uses 111b for bits 7-5.

## Command List

[Table 2](#) provides a listing of the commands and their description.

**Table 2. Command List**

| Address (Hex) | Size (Bytes)      | Direction | Peripheral | Description  |
|---------------|-------------------|-----------|------------|--|
| 0x00          | 1                 | Write     | EEPROM     | Write EEPROM (multiple writes in same packet allowed) <sup>3</sup> |
| 0x01          | 1                 | Read      | EEPROM     | Read EEPROM (multiple reads in same packet allowed)                |
| 0x02          | 2                 | Write     | EEPROM     | Write Current Location Register for EEPROM                         |
| 0x03          | 2                 | Read      | EEPROM     | Read Current Location Register for EEPROM                          |
| 0x04          | 1                 | Write     | EEPROM     | Erase requested Page in EEPROM                                     |
| 0x06          | 2(4) <sup>1</sup> | Write     | GPIO       | Setting GPIO Out Register  |
| 0x07          | 1(2) <sup>1</sup> | Read      | GPIO       | Reading GPIO In Register   |
| 0x08          | 3(5) <sup>2</sup> | Write     | GPIO       | Write GPIO Configuration   |
| 0x09          | 3(5) <sup>2</sup> | Read      | GPIO       | Read GPIO Configuration  |
| 0x0F          | 1                 | Read      | GPIO       | Read GPIO Interrupt Status Register                                |
| 0x21          | 1                 | Read      | UART0      | Read UART Status Register  |
| 0x22          | 1                 | Write     | UART0      | Enable Interrupts  |
| 0x23          | 1                 | Read      | UART0      | Interrupt Status Register  |
| 0x24          | 1                 | Write     | UART0      | Write Data to TX FIFO (multiple bytes allowed) <sup>3</sup>        |
| 0x25          | 1                 | Read      | UART0      | Read RX FIFO Data (Multiple Reads Allowed)                         |
| 0x26          | 2                 | Write     | UART0      | Write Baud Rate Register   |
| 0x27          | 2                 | Read      | UART0      | Read Actual Baud Rate Register                                     |
| 0x28          | 2                 | Write     | UART0      | Write Configuration  |
| 0x29          | 2                 | Read      | UART0      | Read Configuration   |
| 0x2A          | 1                 | Write     | UART0      | Write Transmit Watermark Register                                  |

**Table 2. Command List (Continued)**

| Address (Hex) | Size (Bytes) | Direction | Peripheral | Description   |
|---------------|--------------|-----------|------------|---|
| 0x2B          | 1            | Read      | UART0      | Read Transmit Watermark Register                            |
| 0x2C          | 1            | Write     | UART0      | Write Receive Watermark Register                            |
| 0x2D          | 1            | Read      | UART0      | Read Receive Watermark Register                             |
| 0x2E          | 1            | Write     | UART0      | Enable UART   |
| 0x31          | 2            | Read      | UART0      | Read Receive and Transmit FIFO Level Registers              |
| 0x41          | 1            | Read      | UART1      | Read UART Status Register                                   |
| 0x42          | 1            | Write     | UART1      | Enable Interrupts   |
| 0x43          | 1            | Read      | UART1      | Interrupt Status Register                                   |
| 0x44          | 1            | Write     | UART1      | Write data to TX FIFO (multiple bytes allowed) <sup>3</sup> |
| 0x45          | 1            | Read      | UART1      | Read RX FIFO data (multiple reads allowed)                  |
| 0x46          | 2            | Write     | UART1      | Write Baud Rate Register                                    |
| 0x47          | 2            | Read      | UART1      | Read Actual Baud Rate Register                              |
| 0x48          | 2            | Write     | UART1      | Write Configuration   |
| 0x49          | 2            | Read      | UART1      | Read Configuration  |
| 0x4A          | 1            | Write     | UART1      | Write Transmit Watermark Register                           |
| 0x4B          | 1            | Read      | UART1      | Read Transmit Watermark Register                            |
| 0x4C          | 1            | Write     | UART1      | Write Receive Watermark Register                            |
| 0x4D          | 1            | Read      | UART1      | Read Receive Watermark Register                             |
| 0x4E          | 1            | Write     | UART1      | Enable UART   |
| 0x51          | 2            | Read      | UART1      | Read Receive and Transmit FIFO Level Registers              |
| 0xE1          | 1            | Read      | SYSTEM     | Read System Status Register                                 |
| 0xE3          | 1            | Read      | SYSTEM     | Read Last Operation Result Register                         |
| 0xE5          | 3            | Read      | SYSTEM     | Read System Version   |
| 0xEF          | 1            | Read      | SYSTEM     | Read Interrupt Source Register                              |

Note: 1. ZDU0110QUX device uses 4 bytes, the rest uses 2 bytes.

Note: 2. Command consists of a sub-command and data. The ZDU0110QUX uses 5 bytes, the rest use 3 bytes.

Sub-command 0x0A only uses 1 byte. See Description for more information.

Note: 3. Not allowed to be stacked.

## EEPROM Command Detail

### Write EEPROM at Current Location

**Command:** 0x00

Writes data bytes at the current EEPROM memory location pointer. The bytes will continue to be written in sequential locations until either the maximum bytes have been received (up to 63 bytes) or a stop condition on the I<sup>2</sup>C was received. When the maximum bytes have been received, the Digital UART will NAK the last accepted byte transfer (and any subsequent bytes received). The WP must be low for the EEPROM to actually be written to. After receiving the stop bit on the I<sup>2</sup>C bus, the system does not acknowledge any packet until after the processing of the data is complete. The System Status Register also has the *Busy* bit set until the data has been written to the EEPROM.

### Read EEPROM at Current Location

**Command:** 0x01

Read bytes from the EEPROM at the current EEPROM memory location pointer. The bytes will continue to be sent in sequential locations until either the NAK is received from I<sup>2</sup>C or a Stop condition is sent on the I<sup>2</sup>C bus.

### Write EEPROM Current Location Register

**Command:** 0x02

Sets the current EEPROM memory location pointer to a specific point within the EEPROM's memory

#### Byte 1: High Byte of Address

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0      |
|-------------|-----|-----|-----|-----|-----|-----|-----|--------|
| Description | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Addr 8 |

#### Byte 2: Low Byte of Address

| Bit         | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|-------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Description | Addr 7 | Addr 6 | Addr 5 | Addr 4 | Addr 3 | Addr 2 | Addr 1 | Addr 0 |



## Read EEPROM Current Location Register

**Command:** 0x03

Reads the current EEPROM memory location pointer.

Read Address Format

### Byte 1: High Byte of Address

| Bit         | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
|-------------|---|---|---|---|---|---|---|--------|
| Description | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Addr 8 |

### Byte 2: Low Byte of Address

| Bit         | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|-------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Description | Addr 7 | Addr 6 | Addr 5 | Addr 4 | Addr 3 | Addr 2 | Addr 1 | Addr 0 |

## Erase EEPROM

**Command:** 0x04

Request to erase the EEPROM page number requested in Byte 1 (0-15).

### Byte 1

| Bit         | 7 | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
|-------------|---|---|---|---|--------|--------|--------|--------|
| Description | 0 | 0 | 0 | 0 | Page 3 | Page 2 | Page 1 | Page 0 |

# GPIO Command Detail

## Setting GPIO Out Register

Command: 0x06

Request to set specific GPIO Out pins. The bits set in the MASK byte(s) specify the GPIO pins to be set; all others will be unchanged. If the Bit is set on the Data byte, the GPIO OUT pin will be high, otherwise it will be low (assuming the MASK bit is set). ZDU0110RxX parts only use 1 MASK byte followed by 1 Data byte.

### Byte 1: GPIO Mask High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 2: GPIO Mask Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

### Byte 3: GPIO OUT High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 4: GPIO OUT Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## Reading GPIO In Register

**Command:** 0x07

Reads the current value on the GPIO pins for Input. The ZDU0110QUX device sends two bytes while all other parts only send 1 byte. If the Bit is set, the GPIO pin in question is high.

### Byte 1: GPIO IN High

| Bit                | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|--------------------|-----|-----|-----|-----|--------|--------|-------|-------|
| <b>Description</b> | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| <b>Default</b>     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 2: GPIO IN Low

| Bit                | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>Description</b> | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| <b>Default</b>     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## Write GPIO Configuration

**Command:** 0x08

This command writes specific bits to set configuration for the GPIO pins. The command is followed by a sub command, selecting the configuration in question, the mask to identify the GPIOs to be set, and the values to set. The format for the data being sent is:

Byte 0: 0x08

Byte 1: Sub Command

Byte 2-n: Data for Sub Command

### Sub Commands

0x01 Set GPIO direction (Input/Output)

0x02 Enable GPIO weak Pull ups (Enable/Disable)

0x03 Enable Open Drain Mode (Enable/Disable)

0x04 Enable Debounce (Enable/Disable)

0x05 Enable Interrupt on Pin Change (Enable/Disable)

0x0A Soft Reset of GPIO to return to default settings

## Set GPIO Direction (Input/Output)

**Sub Command:** 0x01

Sets the Input or Output direction of the GPIO pin. The bits set in the MASK byte(s) specify the GPIO pins to be set; all others will be unchanged. If the bit is set on the Data byte, the GPIO will be set to an Output pin. When cleared, the GPIO will be set to an Input pin (assuming the MASK bit is set). ZDU0110RxX parts only use 1 MASK byte followed by 1 Data byte.

### Byte 2: GPIO Mask High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 3: GPIO Mask Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

### Byte 4: GPIO Direction High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 5: GPIO Direction Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## Enable GPIO weak Pull ups (Enable/Disable)

**Sub Command:** 0x02

Enables or disables the weak pull up on the GPIO pin. The bits set in the MASK byte(s) specify the GPIO pins to be set; all others will be unchanged. If the bit is set on the Data byte, the weak pull up resistor on the GPIO pin will be enabled; otherwise, it will be disabled (assuming the MASK bit is set). ZDU0110RxX parts only use 1 MASK byte followed by 1 Data byte.

### Byte 2: GPIO Mask High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 3: GPIO Mask Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

### Byte 4: GPIO Pull Up Enable/Disable High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 5: GPIO Pull Up Enable/Disable Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## Enable Open Drain Mode (Enable/Disable)

**Sub Command:** 0x03

Enables or disables Open Drain Mode for the GPIO pin. The bits set in the MASK byte(s) specify the GPIO pins to be set, all others will be unchanged. If the bit is set on the Data byte, the GPIO pin will be placed in Open Drain (Direction must be set to Output); if the bit is cleared then Open Drain Mode will be disabled (Push-Pull when in Output mode) (assuming the MASK bit is set). ZDU0110RxX parts only use 1 MASK byte followed by 1 Data byte.

### Byte 2: GPIO Mask High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 3: GPIO Mask Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

### Byte 4: GPIO Open Drain Mode Enable/Disable High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 5: GPIO Open Drain Mode Enable/Disable Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## Enable Debounce (Enable/Disable)

**Sub Command:** 0x04

Enables or disables Debounce for the GPIO pin. The bits set in the MASK byte(s) specify the GPIO pins to be set; all others will be unchanged. If the bit is set on the Data byte, the GPIO pin will enable the debounce; if the bit is cleared then debounce will be disabled (assuming the MASK bit is set). ZDU0110RxX parts only use 1 MASK byte followed by 1 Data byte.

### Byte 2: GPIO Mask High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 3: GPIO Mask Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

### Byte 4: GPIO Debounce Enable/Disable High

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

### Byte 5: GPIO Debounce Enable/Disable Low

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## Enable Interrupt on Pin Change (Enable/Disable)

**Sub Command:** 0x05

Enables or disables the Interrupt on Pin value change for GPIO0 and/or GPIO1. The bits set in the MASK byte(s) specifies the GPIO pins to be set; all others will be unchanged. If the bit is set on the Data byte, the GPIO pin will enable the interrupt on the value being toggled, if the bit is cleared then the interrupt will be disabled (assuming the MASK bit is set).

### Byte 2: GPIO Mask

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1     | 0     |
|-------------|-----|-----|-----|-----|-----|-----|-------|-------|
| Description | N/A | N/A | N/A | N/A | N/A | N/A | GPIO1 | GPIO0 |
| Default     | 0   | 0   | 0   | 0   | 0   | 0   | 0     | 0     |

### Byte 3: GPIO Interrupt Enable/Disable

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1     | 0     |
|-------------|-----|-----|-----|-----|-----|-----|-------|-------|
| Description | N/A | N/A | N/A | N/A | N/A | N/A | GPIO1 | GPIO0 |
| Default     | 0   | 0   | 0   | 0   | 0   | 0   | 0     | 0     |

## Soft Reset of GPIO to Return to Default Settings

**Sub Command:** 0x0A

Requests all GPIO pins to return to the default settings (Pull ups, Open Drain, Debounce, and Interrupt all disabled; GPIO pins set to Input).

## Read GPIO Configuration

**Command:** 0x09

Reads the requested configuration for the GPIO pins. The command is followed by a sub command, selecting the configuration to be read.

The data is sent in the following format:

Byte 0: 0x09

Byte 1: Sub Command

Data to be read: ZDU0110QUX: 2 Bytes; ZDU0110RxX: 1 Byte



**Sub Commands:**

- 0x01 GPIO direction (Input/Output)
- 0x02 GPIO weak Pull ups (Enabled/Disabled)
- 0x03 Open Drain Mode (Enabled/Disabled)
- 0x04 Debounce (Enabled/Disabled)
- 0x05 Interrupt on Pin Change (Enabled/Disabled)

**GPIO Direction**

**Sub Command:** 0x01

Reads the current setting for the GPIO direction. If the bit is set, the pin is configured for output; otherwise, it is configured for input.

**Return Byte 1: GPIO Direction High**

| Bit                | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|--------------------|-----|-----|-----|-----|--------|--------|-------|-------|
| <b>Description</b> | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| <b>Default</b>     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

**Return Byte 2: GPIO Direction Low**

| Bit                | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>Description</b> | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| <b>Default</b>     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**GPIO Weak Pull Ups**

**Sub Command:** 0x02

Reads the current setting for the GPIO pull-ups. If the bit is set, the pin has the pull-up resistor enabled; otherwise it is disabled.

**Return Byte 1: GPIO Pull-up High**

| Bit                | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|--------------------|-----|-----|-----|-----|--------|--------|-------|-------|
| <b>Description</b> | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| <b>Default</b>     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

**Return Byte 2: GPIO Pull-up Low**

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**Open Drain Mode**

**Sub Command:** 0x03

Reads the current setting for the GPIO Open Drain Mode. If the bit is set, the GPIO pin is in the Open Drain mode.

**Return Byte 1: GPIO Open Drain Mode High**

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

**Return Byte 2: GPIO Open Drain Mode Low**

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**Debounce**

**Sub Command:** 0x04

Reads the current setting for the GPIO Debounce. If the bit is set, the GPIO pin has the Debounce feature enabled.

**Return Byte 1: GPIO Debounce High**

| Bit         | 7   | 6   | 5   | 4   | 3      | 2      | 1     | 0     |
|-------------|-----|-----|-----|-----|--------|--------|-------|-------|
| Description | N/A | N/A | N/A | N/A | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| Default     | 0   | 0   | 0   | 0   | 0      | 0      | 0     | 0     |

**Return Byte 2: GPIO Debounce Low**

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| Default     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**Interrupt on Pin Change**

**Sub Command:** 0x05

Reads the current setting for interrupt notification on pin change. If the bit is set, any change on the pin will cause the Interrupt line to be asserted.

**Return Byte 1: GPIO Interrupt**

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1     | 0     |
|-------------|-----|-----|-----|-----|-----|-----|-------|-------|
| Description | N/A | N/A | N/A | N/A | N/A | N/A | GPIO1 | GPIO0 |
| Default     | 0   | 0   | 0   | 0   | 0   | 0   | 0     | 0     |

**Read GPIO Interrupt Status Register**

**Command:** 0x0F

Identifies which GPIO interrupted on Port Change. If set, an interrupt occurred; if cleared, there is no interrupt. Register is cleared (and interrupt is released) when read.

**Return Byte 1: GPIO Interrupt**

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1     | 0     |
|-------------|-----|-----|-----|-----|-----|-----|-------|-------|
| Description | N/A | N/A | N/A | N/A | N/A | N/A | GPIO1 | GPIO0 |
| Default     | 0   | 0   | 0   | 0   | 0   | 0   | 0     | 0     |

**UART Command Detail**

**Read UART Status Register**

**Command: UART0:** 0x21; **UART1:** 0x41

Retrieves the status of the UART packets and information. Reading this register will clear the errors (bits 0-3).

| Bit                | 7        | 6       | 5       | 4   | 3     | 2   | 1  | 0  |
|--------------------|----------|---------|---------|-----|-------|-----|----|----|
|                    | TX Empty | TX Full | RX Data | N/A | Break | DOR | FE | PE |
| <b>Description</b> |          |         |         |     |       |     |    |    |

TX Empty: Transmit buffer is empty if set

TX Full: Transmit buffer is full if set

RX Empty: Receive buffer has data if set

Break: Break received

DOR: Data over run error if set

FE: Frame error if set

PE: Parity error if set

## Enable Interrupts

**Command: UART0:** 0x22; **UART1:** 0x42

When the bit is set, the system will assert the Interrupt line when the condition is met.

| Bit                | 7        | 6            | 5       | 4     | 3       | 2            | 1       | 0           |
|--------------------|----------|--------------|---------|-------|---------|--------------|---------|-------------|
|                    | TX Empty | TX Watermark | TK Full | Break | RX Data | TX Watermark | RX Full | TX/RX Error |
| <b>Description</b> |          |              |         |       |         |              |         |             |
| <b>Default</b>     | 0        | 0            | 0       | 0     | 0       | 0            | 0       | 0           |

TX Empty: Transmit buffer is empty if set

TX Full: Transmit buffer is full if set

RX Empty: Receive buffer has data if set

Break: Break received

DOR: Data over run error if set

FE: Frame error if set

PE: Parity error if set

## Interrupt Status Register

**Command: UART0:** 0x23; **UART1:** 0x43

When the bit is set, the condition has been met. Reading this register will clear the TX/RX Error Interrupt.

| Bit                | 7        | 6            | 5       | 4   | 3       | 2            | 1       | 0           |
|--------------------|----------|--------------|---------|-----|---------|--------------|---------|-------------|
| <b>Description</b> | TX Empty | TX Watermark | TK Full | N/A | RX Data | RX Watermark | RX Full | TX/RX Error |

TX Empty: Transmit buffer is Empty

TX Watermark: The watermark for the Transmit has been reached

TX Full: The Transmit buffer is full

RX Data: The Receive Buffer has data

RX Watermark: The Receive Buffer has passed the watermark level

RX Full: The Receive Buffer is full

TX/RX Error: The UART has received an error (check status for which one)

## Write Data to TX FIFO

**Command: UART0:** 0x24; **UART1:** 0x44

Writes all bytes received to TX FIFO until the packet is terminated or TX Buffer is full. I<sup>2</sup>C terminates by Stop condition. If the buffer is full, the I<sup>2</sup>C transmits a Negative Acknowledgement (NAK) for any byte sent until the buffer has been emptied enough to take more data. Any byte that has been NAK'd is not added to the buffer.

### Byte 1 Through Byte n

| Bit                | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>Description</b> | Data7 | Data6 | Data5 | Data4 | Data3 | Data2 | Data1 | Data0 |

## Read RX FIFO Data

**Command: UART0:** 0x25; **UART1:** 0x45

Reads bytes from the RX FIFO until either a Stop Condition or a NAK is received. If the reads exceed the number of bytes in the FIFO, the returning value will be 0x00.

**Byte 1 Through Byte n**

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | Data7 | Data6 | Data5 | Data4 | Data3 | Data2 | Data1 | Data0 |

## Write Baud Rate Register

**Command: UART0:** 0x26; **UART1:** 0x46

Sets the baud rate for the requested UART, in multiples of 100.

---

► **Note:** For more information about the baud rate, see [UART](#) on page 11.

---

**Byte 1: Baud Rate (in 100s) High**

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1  | 0  |
|-------------|-----|-----|-----|-----|-----|-----|----|----|
| Description | B15 | B14 | B13 | B12 | B11 | B10 | B9 | B8 |

**Byte 2: Baud Rate (in 100s) Low**

| Bit         | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|-------------|----|----|----|----|----|----|----|----|
| Description | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

## Read Actual Baud Rate Register

**Command: UART0:** 0x27; **UART1:** 0x47

Reads the actual baud rate in the baud rate register (in multiple of 100s).

---

► **Note:** For more information about the baud rate, see [UART](#) on page 11.

---

**Byte 1: Baud Rate (in 100s) High**

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1  | 0  |
|-------------|-----|-----|-----|-----|-----|-----|----|----|
| Description | B15 | B14 | B13 | B12 | B11 | B10 | B9 | B8 |

**Byte 2: Baud Rate (in 100s) Low**

| Bit         | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
|-------------|----|----|----|----|----|----|----|----|
| Description | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

## Write Configuration

**Command: UART0:** 0x28; **UART1:** 0x48

The configuration for the UART can either be set with all parameters in one command or individual parameters, by the use of sub commands. The first byte, bit 7, determines if this is a sub command request or data (configuring all parameters). If the bit 7 is cleared, the data is a sub command; otherwise, the data is all configuration parameters. The sub commands also provides the ability to reset the FIFO buffers and a soft reset of the UART to return to default settings.

### Setting All Parameters At Once

2 bytes are required to provide all the parameters at once. Bit 7 of Byte 1 **MUST** be set.

**Byte 1: Bit 7 is Set**

| Bit                   | 7   | 6   | 5       | 4       | 3         | 2         | 1         | 0        |
|-----------------------|-----|-----|---------|---------|-----------|-----------|-----------|----------|
| Description           | Cmd | N/A | Parity1 | Parity0 | Databits2 | Databits1 | Databits0 | StopBits |
| <b>Default Config</b> | 1   | 0   | 0       | 0       | 0         | 1         | 1         | 0        |

Parity:

- 00 None
- 11 Odd
- 10 Even

Data Bits:

- 000 5 bit data
- 001 6 bit data
- 010 7 bit data
- 011 8 bit data
- 111 9 bit data

Stop Bit:

- 0 1 stop bit
- 1 2 stop bits

**Byte 2:**

| Bit                | 7   | 6   | 5   | 4   | 3        | 2            | 1            | 0   |
|--------------------|-----|-----|-----|-----|----------|--------------|--------------|-----|
| <b>Description</b> | N/A | N/A | TXE | RXE | Loopback | FlowControl1 | FlowControl0 | N/A |
| <b>Default</b>     | 0   | 0   | 0   | 0   | 0        | 0            | 0            | 0   |

TXE:

- 0 Disable TX
- 1 Enable TX

RXE:

- 0 Disable RX
- 1 Enable RX

Loopback:

- 0 Disable Loopback mode
- 1 Enable Loopback mode

Flow Control:

- 00 None
- 01 Hardware (CTS/RTR)
- 10 Software (Xon/Xoff)

**Setting Individual Parameters**

To set individual parameters for the UART configuration, the system requires two bytes – the sub command byte and the data byte. Bit 7 of the sub command byte **MUST** be cleared.

**Sub Commands:**

0x01 Set databits

0x02 Set Parity

0x03 Set Stop bits

0x04 Set Flow Control

0x05 Reserved

0x06 Reset FIFOs

0x07 Reserved

0x08 Set Loopback

0x0A Soft Reset of UART to return to default settings



### Set Data Bits

Sub Command 0x01

#### Byte 2 – Data Bits

| Bit         | 7   | 6   | 5   | 4   | 3   | 2         | 1         | 0         |
|-------------|-----|-----|-----|-----|-----|-----------|-----------|-----------|
| Description | N/A | N/A | N/A | N/A | N/A | Databits2 | Databits1 | Databits0 |
| Default     | 0   | 0   | 0   | 0   | 0   | 0         | 0         | 0         |

#### Data Bits:

000 5 bit data  
 001 6 bit data  
 010 7 bit data  
 011 8 bit data  
 111 9 bit data

### Set Parity

Sub Command 0x02

#### Byte 2 – Parity

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1       | 0       |
|-------------|-----|-----|-----|-----|-----|-----|---------|---------|
| Description | N/A | N/A | N/A | N/A | N/A | N/A | Parity1 | Parity2 |
| Default     | 0   | 0   | 0   | 0   | 0   | 0   | 0       | 0       |

#### Parity:

00 None  
 11 Odd  
 10 Even

### Set Stop Bits

Sub Command 0x03

#### Byte 2 – Stop Bits

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0        |
|-------------|-----|-----|-----|-----|-----|-----|-----|----------|
| Description | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Stop Bit |
| Default     | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0        |

Stop Bits:  
0 1 stop bit  
1 2 stop bit

### Set Flow Control

Sub Command 0x04

#### Byte 2 – Flow Control

| Bit                | 7   | 6   | 5   | 4   | 3   | 2   | 1            | 0            |
|--------------------|-----|-----|-----|-----|-----|-----|--------------|--------------|
| <b>Description</b> | N/A | N/A | N/A | N/A | N/A | N/A | FlowControl1 | FlowControl2 |
| <b>Default</b>     | 0   | 0   | 0   | 0   | 0   | 0   | 0            | 0            |

Flow Control:  
00 None  
01 Hardware (CTS/RTR)  
10 Software (Xon/Xoff)

### Reset FIFOs

Sub Command 0x06

#### Byte 2 – Reset FIFOs

| Bit                | 7   | 6   | 5   | 4   | 3   | 2   | 1       | 0       |
|--------------------|-----|-----|-----|-----|-----|-----|---------|---------|
| <b>Description</b> | N/A | N/A | N/A | N/A | N/A | N/A | RX FIFO | TX FIFO |
| <b>Default</b>     | 0   | 0   | 0   | 0   | 0   | 0   | 0       | 0       |

Reset FIFO:  
TXFIFO:  
0 Do not reset  
1 Reset TX FIFO  
RXFIFO:  
0 Do not reset  
1 Reset RX FIFO

### Set Loopback

Sub Command 0x08

**Byte 2 – Loopback**

| Bit         | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0        |
|-------------|-----|-----|-----|-----|-----|-----|-----|----------|
| Description | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Loopback |
| Default     | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0        |

Loopback:

0 - Loopback Off (normal operations)

1 - Loopback On

**Reset UART**

Sub Command 0x0A

This command does a soft reset for the UART and returns back to default conditions. No data is to be sent with this sub command.

**Read Configuration**

**Command: UART0:** 0x29; **UART1:** 0x49

Reads the current configuration of the UART

**Byte 1 Returned:**

| Bit         | 7   | 6   | 5       | 4       | 3         | 2         | 1         | 0        |
|-------------|-----|-----|---------|---------|-----------|-----------|-----------|----------|
| Description | N/A | N/A | Parity1 | Parity0 | Databits2 | Databits1 | Databits0 | StopBits |
| Default     | 0   | 0   | 0       | 0       | 0         | 1         | 1         | 0        |

Parity:

00 None

11 Odd

10 Even

Data Bits:

000 5 bit data

001 6 bit data

010 7 bit data

011 8 bit data

111 9 bit data

Stop bit:

- 0 1 stop bit
- 1 2 stop bits

**Byte 2 Returned:**

| Bit                | 7   | 6   | 5   | 4   | 3        | 2            | 1            | 0   |
|--------------------|-----|-----|-----|-----|----------|--------------|--------------|-----|
| <b>Description</b> | N/A | N/A | TXE | RXE | Loopback | FlowControl1 | FlowControl0 | N/A |
| <b>Default</b>     | 0   | 0   | 0   | 0   | 0        | 0            | 0            | 0   |

**TXE:**

- 0 Disable TX
- 1 Enable TX

**RXE:**

- 0 Disable RX
- 1 Enable RX

**Loopback:**

- 0 Disable Loopback mode
- 1 Enable Loopback mode

**Flow Control:**

- 00 None
- 01 Hardware (CTS/RTR)
- 10 Software (Xon/Xoff)

## Write Transmit Watermark Register

**Command: UART0:** 0x2A; **UART1:** 0x4A

Writes the Transmit Watermark Register.

Values between 1 and 64 are acceptable. Sending a 0x00 will keep the TX Watermark Interrupt always on and any value greater than 64 will not ever set the Interrupt. The default value is 63.

## Read Transmit Watermark Register

**Command: UART0:** 0x2B; **UART1:** 0x4B

Reads the Transmit Watermark Register.

## Write Receive Watermark Register

**Command: UART0:** 0x2C; **UART1:** 0x4C

Writes the Receive Watermark Register.

Values between 1 and 64 are acceptable. Sending a 0x00 value will keep the RX Watermark Interrupt always on and any value greater than 64 will never set the Interrupt. The default value is 1.

## Read Receive Watermark Register

**Command: UART0:** 0x2D; **UART1:** 0x4D

Reads the Receive Watermark Register.

## Enable UART

**Command: UART0:** 0x2E; **UART1:** 0x4E

Enables the UART for receiving and/or transmitting.

### Byte 1: Enable UART RX and TX

| Bit                | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| <b>Description</b> | N/A | N/A | N/A | N/A | N/A | N/A | TXE | RXE |

TXE:

- 0 Disable TX
- 1 Enable TX

RXE:

- 0 Disable RX
- 1 Enable RX

## Read Receive and Transmit FIFO Level Registers

**Command: UART0:** 0x31; **UART1:** 0x51

Retrieves the current FIFO level for the Receive and Transmit FIFOs.

The maximum value is 64 and the minimum value is 0.

### Byte 1: RX FIFO Level

| Bit                | 7   | 6   | 5      | 4      | 3      | 2      | 1      | 0      |
|--------------------|-----|-----|--------|--------|--------|--------|--------|--------|
| <b>Description</b> | N/A | N/A | Level5 | Level4 | Level3 | Level2 | Level1 | Level0 |
| <b>Default</b>     | 0   | 0   | 0      | 0      | 0      | 0      | 0      | 0      |

**Byte 2: TX FIFO Level**

| Bit                | 7   | 6   | 5      | 4      | 3      | 2      | 1      | 0      |
|--------------------|-----|-----|--------|--------|--------|--------|--------|--------|
| <b>Description</b> | N/A | N/A | Level5 | Level4 | Level3 | Level2 | Level1 | Level0 |
| <b>Default</b>     | 0   | 0   | 0      | 0      | 0      | 0      | 0      | 0      |

## System Command Detail

### Read System Status Register

**Command:** 0xE1

Provides the status of the system.

**Byte 1: Status**

| Bit                | 7    | 6   | 5   | 4   | 3   | 2   | 1        | 0    |
|--------------------|------|-----|-----|-----|-----|-----|----------|------|
| <b>Description</b> | Init | N/A | N/A | N/A | N/A | N/A | Op Error | Busy |

Init: When set, the system is being initialized. When cleared, the system is operational.

Busy: When set, the system is busy processing a request.

Op Error: When not busy, sets the bit if there was an error on the last register write operation performed.

---

► **Note:** If the system is busy processing a write request, the system will NAK any request, including the System Status Register, until the write request is completed.

---

### Read Last Operation Result Register

**Command:** 0xE3

This command provides the operational result for the last command request. A 0x00 value indicates success and a non-zero value indicates an error.

### Read System Version

**Command:** 0xE5

Reads the version of the Digital UART part.

**Byte 1: Part Identifier (1-99)**

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |

**Byte 2: Major Revision (1-99)**

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |

**Byte 1: Minor Revision (0-99)**

| Bit         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Description | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |

## Read Interrupt Source Register

**Command:** 0xEF

Provides the source of any active interrupt. Used to quickly identify which peripheral has requested the interrupt when there is more than one peripheral on a single interrupt line, such as the UART.

**Byte 1: Source**

| Bit         | 7   | 6   | 5   | 4   | 3   | 2  | 1  | 0    |
|-------------|-----|-----|-----|-----|-----|----|----|------|
| Description | N/A | N/A | N/A | N/A | N/A | U1 | U0 | GPIO |
| Default     | 0   | 0   | 0   | 0   | 0   | 0  | 0  | 0    |

U1: External UART1 - Interrupt on the UART1 (if available)

U0: External UART0 - Interrupt on the UART0

GPIO: GPIO – Interrupt on GPIO pin toggle

# Electrical Characteristics

This chapter lists the electrical characteristics of the Digital UART.

## Absolute Maximum Ratings

| Parameter          | Symbol                  | Rating         | Unit     |    |
|--------------------|-------------------------|----------------|----------|----|
| Supply Voltage     | VDD                     | -0.3 ~ +6.5    | V        |    |
|                    | VSS                     | -0.3 ~ +0.3    | V        |    |
| Normal Voltage Pin | VI                      | -0.3 ~ VDD+0.3 | V        |    |
|                    | VO                      | -0.3 ~ VDD+0.3 | V        |    |
|                    | IOH                     | 10             | mA       |    |
|                    | $\Sigma$ IOH            | 80             | mA       |    |
|                    | IOL                     | 20             | mA       |    |
|                    | $\Sigma$ IOL            | 160            | mA       |    |
|                    | Total Power Dissipation | PT             | 600      | mW |
|                    | Storage Temperature     | TSTG           | -45~+125 | °C |

Note: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.



## Recommended Operating Conditions

| Parameter             | Symbol | Condition    | Minimum | Typical | Maximum | Unit |
|-----------------------|--------|--------------|---------|---------|---------|------|
| Supply Voltage        | VDD    |              | 1.8     |         | 5.5     | V    |
| Operating Temperature | TOPR   | VDD=1.8~5.5V | -40     |         | 85      | °C   |
| Operating Frequency   |        |              | 7.76    |         | 8.24    | MHz  |
|                       |        |              |         | 1       |         | MHz  |

## Power-on Reset Characteristics

| Parameter                    | Symbol | Condition | Minimum | Typical | Maximum | Unit |
|------------------------------|--------|-----------|---------|---------|---------|------|
| Operating Voltage            |        | -         | VSS     | -       | 5.5     | V    |
| Operating Temperature        |        | -         | -40     |         | 85      | °C   |
| RESET Release Level          |        | -         | 1.3     | 1.4     | 1.5     | V    |
| nRESET Input Pulse "L" Width | tRST   | VDD=5.0V  | 1.1     | -       | -       | µs   |
| Operating Current            | IDD    | -         | -       | -       | 10      | µA   |
|                              | SIDD   | -         | -       | -       | 1       | µA   |

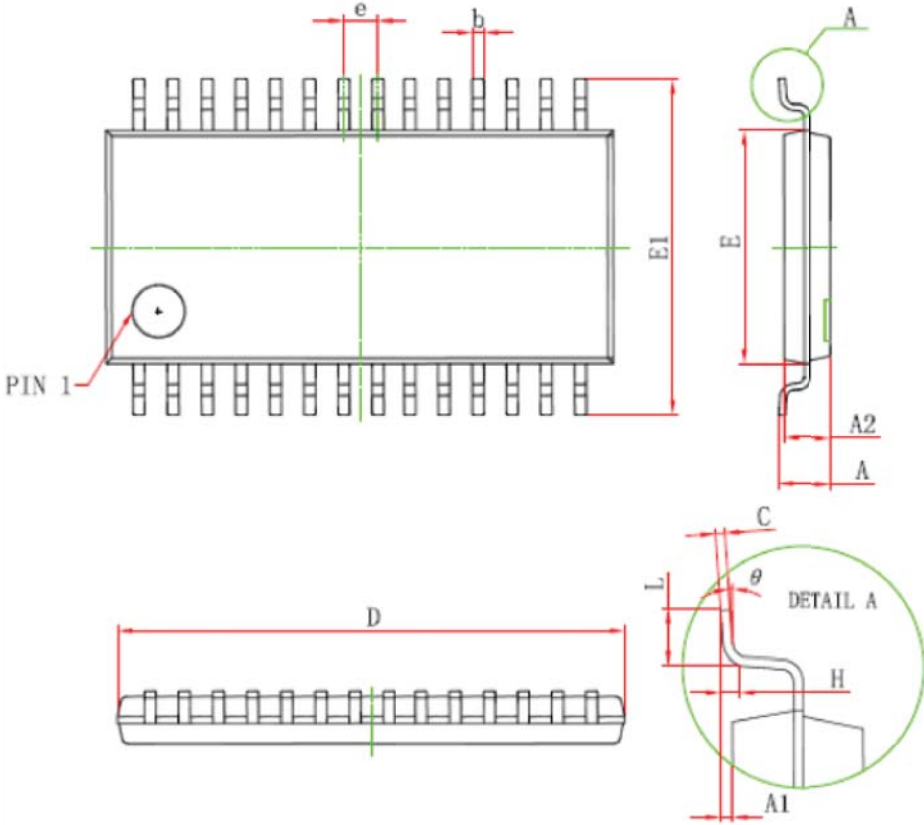
## Internal RC Oscillator Characteristics

| Parameter             | Symbol | Condition | Minimum | Typical | Maximum | Unit |
|-----------------------|--------|-----------|---------|---------|---------|------|
| Operating Voltage     |        | –         | 1.8     | –       | 5.5     | V    |
| Operating Temperature |        | –         | –40     |         | 85      | °C   |
| Frequency             |        | –         | 7.76    |         | 8.24    | MHz  |
| Stabilization Time    |        | –         | –       | –       | 10      | mS   |
| Operating Current     | IDD    | –         | –       | –       | –       | uA   |
|                       | SIDD   | –         | –       | –       | 1       | uA   |

## DC Characteristics

| Parameter                  | Symbol | Condition                   | Minimum | Typical | Maximum | Unit |
|----------------------------|--------|-----------------------------|---------|---------|---------|------|
| Input Low Voltage          | VIL    | –                           | –0.5    | –       | 0.2VDD  | V    |
| Input High Voltage         | VIH    | –                           | 0.7VDD  | –       | VDD     | V    |
| Output Low Voltage         | VOL    | IOL=20mA,<br>Vdd=4.5V       | –       | –       | 1       | V    |
| Output High Voltage        | VOH    | IOH=8.57mA,<br>VDD=4.5V     | 3.5     | –       | –       | V    |
| Input High Leakage Current | IIH    |                             |         |         | 1       | uA   |
| Input Low Leakage Current  | IIL    |                             | –1      |         |         | uA   |
| Pull-up Resistor           | RPU    |                             | 20      |         | 50      | kΩ   |
| Power Supply Current       | IDD    | VDD=5V, All<br>GPIO OUT low | –       | 2.6     | –       | mA   |

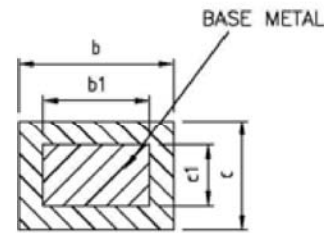
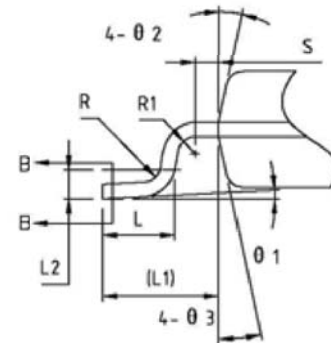
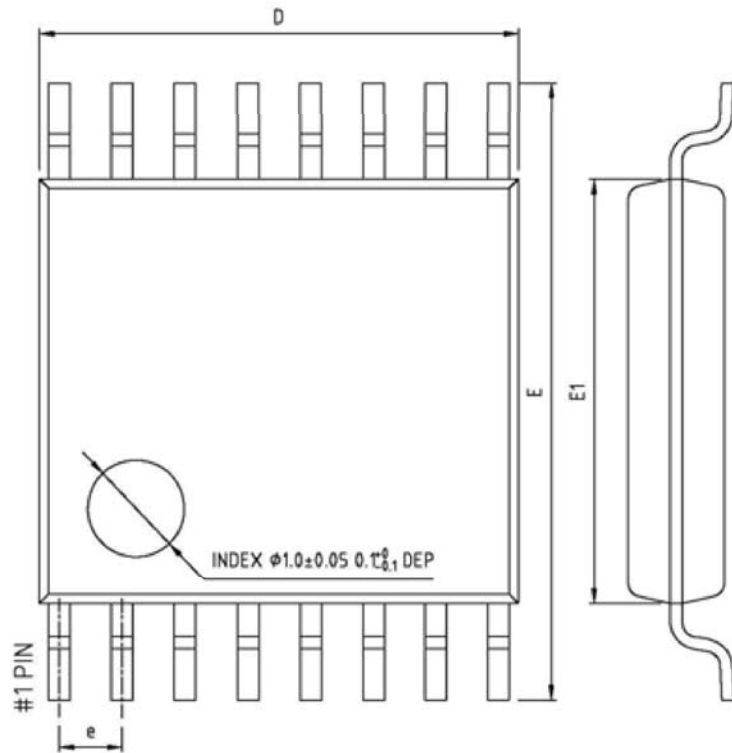




| Symbol | Dimensions In Millimeters |       | Dimensions In Inches |       |
|--------|---------------------------|-------|----------------------|-------|
|        | Min                       | Max   | Min                  | Max   |
| D      | 9.600                     | 9.800 | 0.378                | 0.386 |
| E      | 4.300                     | 4.500 | 0.169                | 0.177 |
| b      | 0.190                     | 0.300 | 0.007                | 0.012 |
| c      | 0.090                     | 0.200 | 0.004                | 0.008 |
| E1     | 6.250                     | 6.550 | 0.246                | 0.258 |
| A      |                           | 1.200 |                      | 0.047 |
| A2     | 0.800                     | 1.000 | 0.031                | 0.039 |
| A1     | 0.050                     | 0.150 | 0.002                | 0.006 |
| e      | 0.65 (BSC)                |       | 0.026 (BSC)          |       |
| L      | 0.500                     | 0.700 | 0.020                | 0.028 |
| H      | 0.25(TYP)                 |       | 0.01(TYP)            |       |
| θ      | 1°                        | 7°    | 1°                   | 7°    |

Figure 7. 28-Pin TSSOP Package

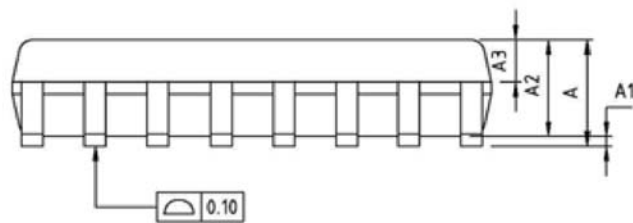
Figure 8.20-Pin TSSOP Package



SECTION B-B

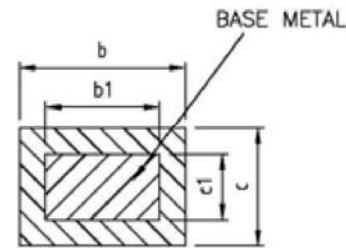
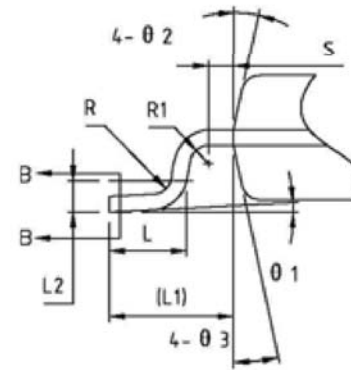
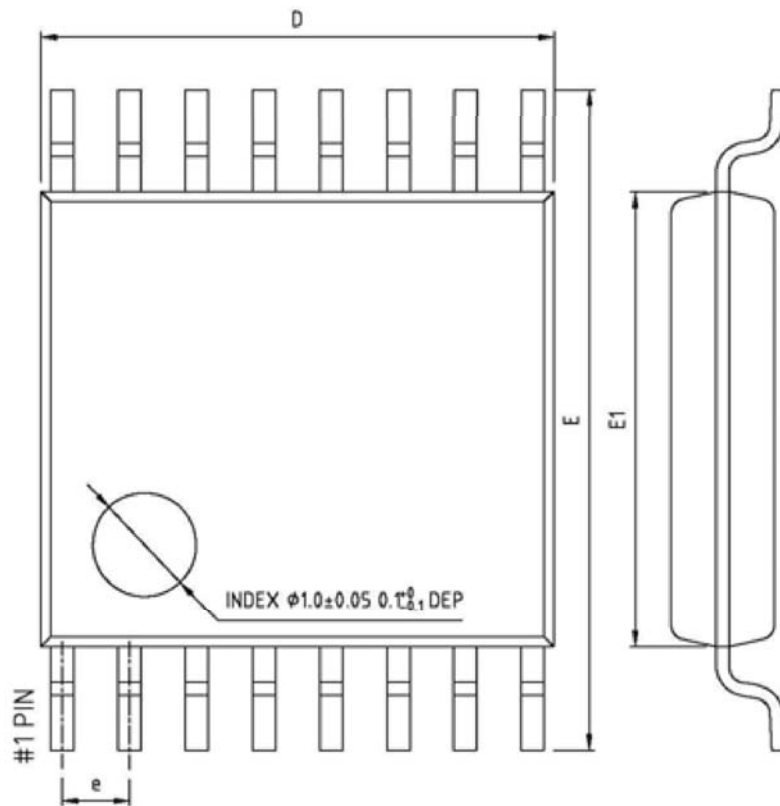
COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

| SYMBOL         | MIN     | NOM  | MAX  |
|----------------|---------|------|------|
| A              | —       | —    | 1.20 |
| A1             | 0.05    | —    | 0.15 |
| A2             | 0.90    | 1.00 | 1.05 |
| A3             | 0.34    | 0.44 | 0.54 |
| b              | 0.20    | —    | 0.28 |
| b1             | 0.20    | 0.22 | 0.24 |
| c              | 0.10    | —    | 0.19 |
| c1             | 0.10    | 0.13 | 0.15 |
| D              | 4.86    | 4.96 | 5.06 |
| E              | 6.20    | 6.40 | 6.60 |
| E1             | 4.30    | 4.40 | 4.50 |
| e              | 0.65BSC |      |      |
| L              | 0.45    | 0.60 | 0.75 |
| L1             | 1.00REF |      |      |
| L2             | 0.25BSC |      |      |
| R              | 0.09    | —    | —    |
| R1             | 0.09    | —    | —    |
| S              | 0.20    | —    | —    |
| θ <sub>1</sub> | 0°      | —    | 8°   |
| θ <sub>2</sub> | 10°     | 12°  | 14°  |
| θ <sub>3</sub> | 10°     | 12°  | 14°  |



NOTES:  
ALL DIMENSIONS REFER TO JEDEC STANDARD MO-153 AB  
DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

Figure 9.16-Pin TSSOP Package



SECTION B-B

COMMON DIMENSIONS  
(UNITS OF MEASURE=MILLIMETER)

| SYMBOL | MIN     | NOM  | MAX  |
|--------|---------|------|------|
| A      | —       | —    | 1.20 |
| A1     | 0.05    | —    | 0.15 |
| A2     | 0.90    | 1.00 | 1.05 |
| A3     | 0.34    | 0.44 | 0.54 |
| b      | 0.20    | —    | 0.28 |
| b1     | 0.20    | 0.22 | 0.24 |
| c      | 0.10    | —    | 0.19 |
| c1     | 0.10    | 0.13 | 0.15 |
| D      | 4.86    | 4.96 | 5.06 |
| E      | 6.20    | 6.40 | 6.60 |
| E1     | 4.30    | 4.40 | 4.50 |
| e      | 0.65BSC |      |      |
| L      | 0.45    | 0.60 | 0.75 |
| L1     | 1.00REF |      |      |
| L2     | 0.25BSC |      |      |
| R      | 0.09    | —    | —    |
| R1     | 0.09    | —    | —    |
| S      | 0.20    | —    | —    |
| θ1     | 0°      | —    | 8°   |
| θ2     | 10°     | 12°  | 14°  |
| θ3     | 10°     | 12°  | 14°  |



NOTES:  
ALL DIMENSIONS REFER TO JEDEC STANDARD MO-153 AB  
DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

# Ordering Information

[Table 3](#) identifies the basic features and package styles available for the Digital UART.

**Table 3. Ordering Information**

| Part Number | UARTs | GPIOs | Package      |
|-------------|-------|-------|--------------|
| ZDU0110RFX  | 1     | 3     | 16 pin TSSOP |
| ZDU0110RHX  | 1     | 7     | 20 pin TSSOP |
| ZDU0210RJX  | 2     | 8     | 28 pin TSSOP |
| ZDU0210QUX  | 2     | 12    | 32 pin QFN   |

## *Customer Support*

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at <http://support.zilog.com>.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at <http://zilog.com/kb> or consider participating in the Zilog Forum at <http://zilog.com/forum>.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at <http://www.zilog.com>.