Uart Implementation Reference

v1.0.0

Written by
Kevin LASTRA

Contents

1 Introduction

1.1 Purpose of the manual

The purpose of this manual is twofold. First, it serves as a comprehensive resource to document the Implementation of a Universal Asynchronous Receiver-Transmitter (UART). This process has been undertaken not only as a journey of self-learning but also as a way to share knowledge and contribute to the broader community of developers, engineers, and enthusiasts.

1.2 Audience

This manual is primarily intended for design engineers who are involved in the development and implementation of communication systems, particularly those working with embedded systems, microcontrollers, and hardware interfaces.

2 UART overview

The Universal Asynchronous Receiver-Transmitter (UART) is a fundamental hardware communication protocol used for asynchronous serial communication.
This protocol is widely used in embedded systems, microcontrollers, and communication devices due to its simplicity and effectiveness in handling relatively low-speed data transfer.

2.1 UART fundamentals

2.1.1 Reception and transmission

UART allows two devices to exchange data using only two wires, RX for receive data and TX for transmit data.

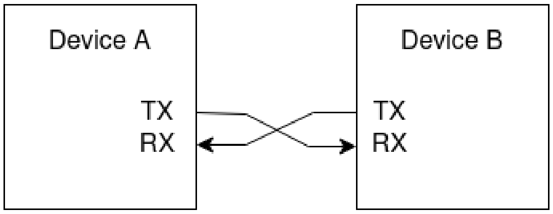

Figure 1: Two UART's interfaces connected

2.1.2 Asynchronous communication

An asynchronous communication refers to a type of data transmission where the sender and receiver do not rely on a shared clock signal for synchronization. Instead, each device operates using its own internal clock and relies on specific timing conventions to ensure data is transmitted and received correctly.

In asynchronous UART communication, the data is sent in discrete chunks (called data frames) over the communication channel, with the timing governed by agreed-upon parameters like baud rate and frame size.

2.1.3 Baud rate

The baud rate defines the speed at which data is transmitted over the communication channel. It is usually expressed in bits per second (bps).
In the context of UART the baud rate specifies how many bits of data can be transmitted each second. Both transmitter and receiver must be set to the same baud rate.

The general formula for calculating the baud rate is:

$$BaudRate = \frac{ClockFrequency}{Divisor}$$

2.1.4 Data frame structure

A UART data frame typically consists of:

Range name	**Description**	**Implementation bit length**
Start bit	Signals the beginning of data transmission	1
Data bits	The actual data being sent	5 - 9
Parity	Used for error checking	0 - 1
Stop bit	Signals the end of the data transmission	1 - 2

2.2 Advantages and limitations

Advantages:

1. Minimal pins required, essay to implement on a system.
2. Low cost, because of its minimal hardware requirements the UART is a cost-effective solution.
3. Widely supported.
4. Simplicity in software implementation.

limitations:

1. Distance limitations, the maximum range depends on the baud rate, the quality of the wires and the environment (e.g. electromagnetic interferences).

2. Relatively slow data transfer.

3. Limited to two devices.

4. Susceptibility to baud rate mismatch.

5. Limited error detection, UART error detection is limited to parity checks and stop bits.

3 Implementation

3.1 Design

3.2 Interface

3.3 Registers

3.4 Interrupt request (IRQ)

3.5 Low power