Software Requirements Specification (SRS)

UART

Authors: Burners Team (Ahmed Salah, Basem Moufreh, Hassan El Gabass, Hazem El Morshedi, Mohamed Safwat).

Customer: NTI

Instructor: Mahmoud Ali, Ahmed Abd El Reheem.

Contents

National Telecommunication Institute

1 Introduction

The UART (Universal Asynchronous Receiver-Transmitter) module is a software component that enables serial communication between a microcontroller and other devices. This document outlines the requirements for the development and implementation of the UART module.

a. Purpose: The purpose of this document is to define the requirements for the software system that interacts with the UART communication protocol.

b. Scope: The UART module will provide the necessary functionality to establish serial communication using the UART protocol. It will handle data transmission and reception, support configurable baud rates, and provide error detection and handling mechanisms.

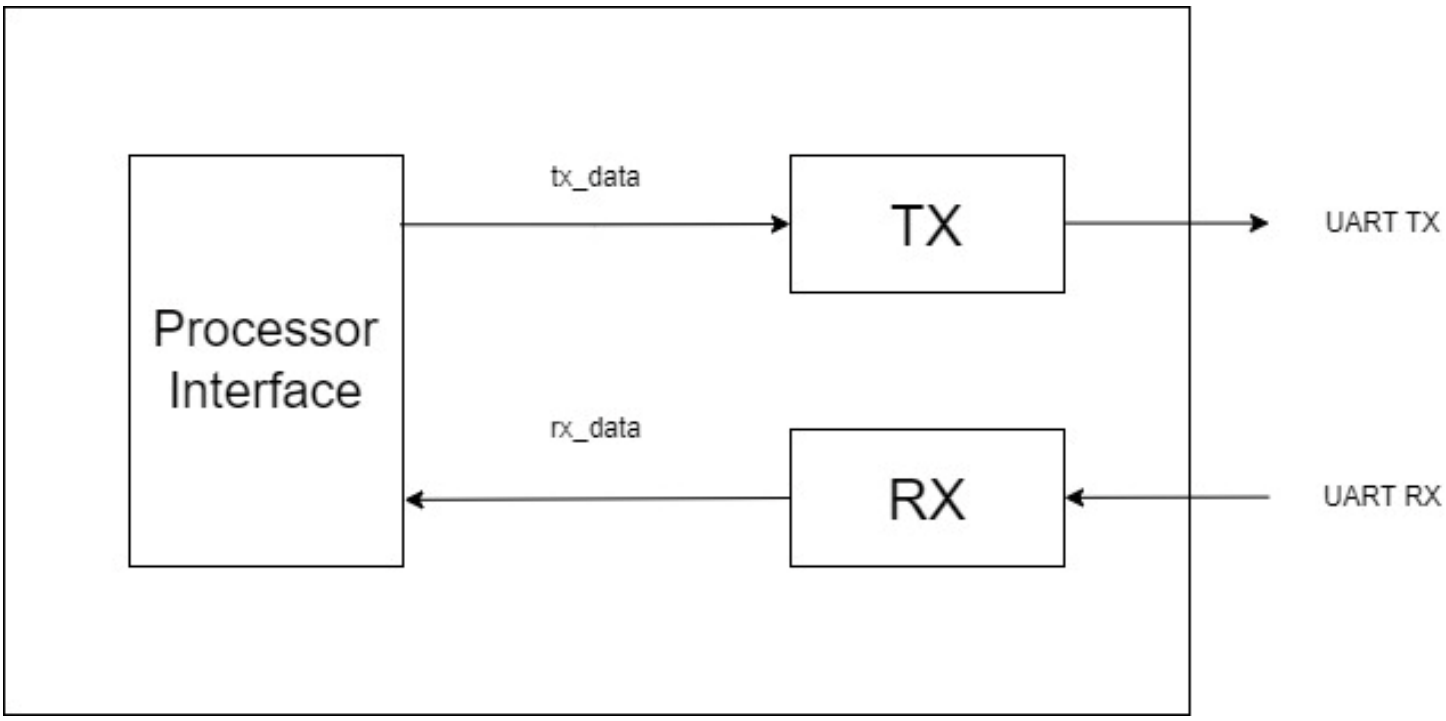
c. Definitions, Acronyms, and Abbreviations:

UART: Universal Asynchronous Receiver Transmitter.

2 System Overview

The UART (Universal Asynchronous Receiver-Transmitter) module is a software component that facilitates serial communication between a microcontroller and other devices. It provides the necessary functionality to transmit and receive asynchronous serial data using the UART protocol. The system overview of the UART module includes its key components, interactions, and the overall flow of data within the system.

Context diagram



3 Functional Requirements

[SRS_UART_100]: UART Initialization

The module shall provide an initialization function to set up the UART communication. The function shall configure the UART hardware, including baud rate, data bits, parity, and stop bits. The function shall set up interrupt handlers for transmit and receive events.

[SRS_UART_101]: Data Transmission

The module shall provide a function to send data through the UART.

The function shall accept a buffer of data and its length as parameters.

The function shall transmit the data synchronously or asynchronously, based on the configuration.

The module shall handle any errors that occur during transmission, such as buffer overflow or transmission failure.

[SRS_UART_102]: Data Reception

The module shall provide a function to receive data from the UART.

The function shall accept a buffer and its length as parameters.

The module shall receive data synchronously or asynchronously, based on the configuration.

The function shall return the number of bytes received.

The module shall handle any errors that occur during reception, such as buffer overflow or reception failure.

[SRS_UART_103]: Interrupt Handling

The module shall support interrupt-driven communication for both transmission and reception.

The module shall trigger the appropriate interrupt handlers upon events like transmit complete or data received.

The interrupt handlers shall handle the corresponding events and perform the necessary actions.

[SRS_UART_104]: Baud Rate Configuration

The module shall provide a function to configure the baud rate.

The function shall accept the desired baud rate as a parameter.

The module shall calculate and set the necessary registers to achieve the specified baud rate.

[SRS_UART_105]: Error Detection and Handling

The module shall implement error detection mechanisms, such as parity or checksum, to ensure data integrity.

The module shall provide error handling capabilities, such as error flags or callbacks, to notify the application of any errors.

4 Non-Functional Requirements

[SRS_UART_120]: Performance

The UART module shall have low latency and provide efficient data transmission and reception. The module shall be capable of handling large volumes of data without data loss or corruption.

[SRS_UART_120]: Reliability

The module shall be robust and reliable, capable of operating under various environmental conditions. The module shall handle exceptions and errors gracefully, minimizing the impact on the overall system.

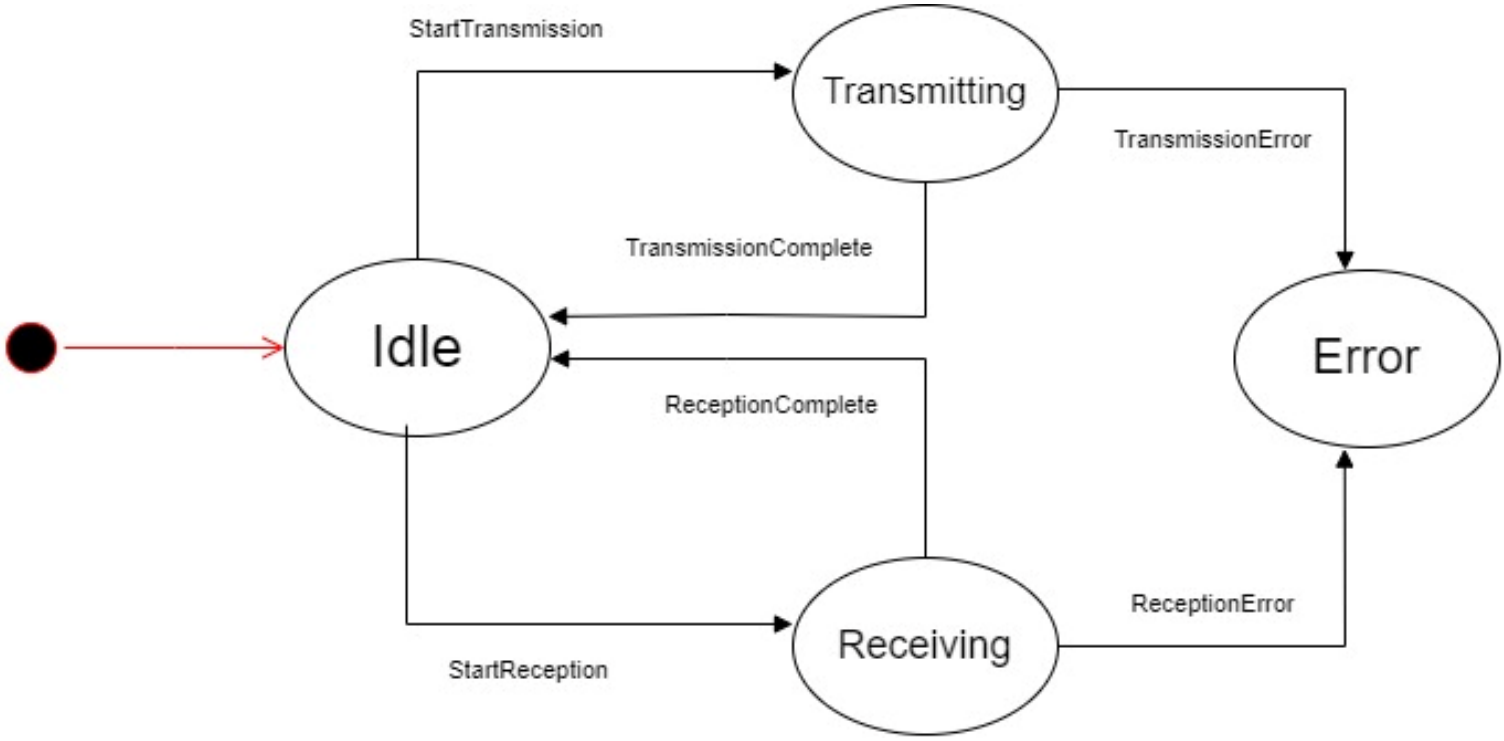
[SRS_UART_120]: Portability

The UART module shall be designed to be portable across different microcontroller platforms and architectures. The module shall be easily adaptable to different operating systems and development environments.

[SRS_UART_120]: Security

The module shall implement security measures, such as data encryption or authentication, if required by the application.

5 State machine



State Descriptions:

1. Idle:

- Initial state of the UART module.
- Waits for a request to transmit or receive data.

2. Transmitting:

- Enters this state upon receiving a StartTransmission request.
- Transmits the data from the transmit buffer.
- Monitors the status of the transmission.
- Transitions to Idle upon successful completion of transmission.
- Transitions to Error if an error occurs during transmission.

3. Receiving:

- Enters this state upon receiving a StartReception request.
- Waits for incoming data from the UART hardware.
- Stores the received data in the receive buffer.
- Transitions to Idle upon successful completion of reception.
- Transitions to Error if an error occurs during reception.

4. Error:

- Enters this state if an error occurs during transmission or reception.
- Sets appropriate error flags or invokes error callbacks.
- Waits for the error to be handled by the application software.
- Transitions to Idle once the error has been processed.

6 Sequence diagram