2024

SPI-USART 2-side gateway

April 2024

Table of Contents

Project was developed for stm32f103c8t6 MCU on "blue pill" board.

April 2024

Requirements for SPI-USART 2-Side Gateway Implementation

In this documentation, we outline the necessary specifications for developing an SPI-USART 2-side gateway. The device is required to operate in master mode, facilitating communication with both USART and SPI devices interchangeably. The system must be designed to seamlessly handle incoming messages from either USART or SPI devices at any given time. To achieve this, two distinct tasks need to be implemented for SPI and USART communication protocols, respectively.

Key Specifications:

Master Mode Operation: The gateway device must function exclusively in master mode to control communication flow effectively.

Dual Protocol Support: The system should be capable of receiving and transmitting data over both SPI and USART interfaces.

Message Handling: Messages received from slave devices should adhere to the following guidelines:

Each frame within a message must be 16 bits in length.

Slave devices are expected to transmit data when available. If no data is available, slave devices will transmit NULL (for SPI) or 0 (for USART), indicating the absence of data.

A single message can encompass up to 10 frames, providing flexibility in data transmission.

Communication Speed: While the desired speed of communication is not explicitly specified, the gateway device should be designed to accommodate various communication speeds efficiently.

Task Design for SPI-USART 2-Side Gateway Implementation

In the development of the SPI-USART 2-side gateway, a meticulous task allocation strategy is essential to ensure efficient data exchange while minimizing processor overhead. Below, we outline the proposed task architecture:

Task 1: USART Data Reception via DMA

This task is dedicated to receiving data from the USART interface utilizing Direct Memory Access (DMA) to offload core processing.

Upon receiving each byte, a timer is initiated for a duration of 100 microseconds within the DMA interrupt handler. This timer serves the crucial function of determining the continuity of the incoming message.

The timer expiration, set at 100 microseconds, acts as a threshold to detect the absence of further bytes in the ongoing message. If no additional bytes are received during this period, it indicates the completion of the message.

The task is engineered to accommodate messages ranging from 1 to 20 bytes in length, corresponding to 1 to 10 frames (where each frame consists of 2 bytes).

Once a complete message is assembled, the bit order is adjusted to align with the desired endianness for SPI transmission (big-endian).

Task 2: SPI Data Reception and Transmission via DMA

Similar to Task 1, this task leverages DMA for both receiving data from SPI and transmitting processed messages back over the SPI interface.

However, due to the inherently slower transmission speed of SPI (125 kHz), the timeout for the timer is adjusted to 70 microseconds, ensuring timely detection of message completion.

Upon receiving a complete message, the bit order is appropriately adjusted to conform to SPI's big-endian byte order before transmission.

Task 3: Low Energy Mode Management

The third task is dedicated to managing the gateway device's power consumption by orchestrating transitions into low-energy states when idle.

This task plays a pivotal role in optimizing power utilization, thereby enhancing the device's overall energy efficiency and prolonging battery life.

By delineating the gateway's functionality into these distinct tasks, we ensure a robust and streamlined implementation capable of efficiently handling data exchange between SPI and USART interfaces while also prioritizing power efficiency through low-energy mode management.