Embedded Systems

Exercise 6

Introduction

This task aims to write an application using UART communication interface that allows serial data transmission between the STM microcontroller and PC. The program should configure the LPUART peripheral and use it for sending and receiving the data. The program must be written in C using STM documentation for the microcontroller registers.

Tasks

- Create a C structure describing registers of the LPUART peripheral device
- Create the library with the following functions required for the LPUART interface:
  - `int LPUART_init(void)` – for configuration and initialisation
  - `int LPUART_SendChar(unsigned char data)` – for sending single characters
  - `int LPUART_ReceiveChar(unsigned char* data)` – for receiving single characters
  - `int LPUART_SendString(unsigned char* data)` – for sending strings
- Write an application according to the requirements specified below.

Initial Questions

As a preparation for writing the program, read chapters 8 (GPIO) and 41 (LPUART) of the STM32L4 reference manual and answer the following questions:

1. How to configure the GPIO port to an appropriate alternate function for the LPUART interface? Which register should be used for configuration and what are the required settings?
   To do so we need to set in port C bits of AFRL register to AF8 configuration which will effectively set pins 3 and 7. In port C we also need to set pins 0 and 1 to AF mode (modification of MODER to 10).
2. Which registers should be used to configure the transmission parameters (character length, number of stop bits)?
   LPUART_CR1 bit M1 for word length: 00 for 8 bit
   LPUART_CR2 bit 12-13 are STOP bits used to determine no of stop bits. 00 for 1.
3. Which register should be used to configure the baud rate?
   LPUART_BRR
4. How to enable the LPUART transmitter? Please specify the register name and bit.
   In register LPUART_CR1 bit UE should be set to 1.
5. How to enable the LPUART receiver? Please specify the register name and bit.
   LPUART_CR1 set bit RE
6. Which register should be used for writing data for sending?
   LPUART_TDR - transmit data register
7. Which register should be used for reading the received data?
   LPUART_RDR - read data register
8. Which register contains information about the status of the LPUART interface and transmission?
   LPUART_ISR - interrupt & status register

Requirements for the application

The final goal of the exercise is to implement an application that can communicate over the LPUART interface with a PC. The application should implement the following functions:

1) At the startup, the application sends an alphabet using small and capital letters using the function `LPUART_SendChar()`, e.g.:
   `"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"`
2) After the alphabet, the application sends an example string using the function `LPUART_SendString()`, e.g.:
   `"Welcome message - <your name>"`
3) Then, the application runs the infinity loop that receives the character from PC, changes its typeface, and sends it back to the PC. Changing the typeface means changing small letters to capital ones and vice versa. Numbers and special characters should be sent back without any modifications.
   Example:
   - Received characters: `asDFgh123`
   - Characters send back: `ASdfGH123`

Additional information

- Communication should be configured to 8N1 mode (8 data bits, no parity, 1 stop bit).
- The baud rate should be configured to 115200
- PuTTY is a console application that can be used on the PC to communicate over the UART interface