UART Tx and Rx Code in Keil

20 June 2025 17:12

```
#include "stm32f429xx.h"


void delay()
{
            for(volatile int i=5000000;i--; );
}
void usart_init()
{
      //Enable gpio clk for port A (usart1), connected to AHB1 bus
      RCC->AHB1ENR |=RCC_AHB1ENR_GPIOAEN;
      //The USART1 is connected to the AP2 bus
      RCC->APB2ENR |= RCC_APB2ENR_USART1EN;

       //Configure Gpio ports
  GPIOA->MODER &=~((3<<(9*2)) | (3<<(10*2)));//clearing the 9th and 10th pin
       GPIOA->MODER |=((2<<(9*2)) | (2<<(10*2)));//setting the pins as AF
       // here we are using 3 (0b11) to clear the 2 bits in each pin
       // here we are using 2 (0b10) to set the alternate function mode in the pins

      GPIOA->OTYPER |=0x00000000;//setting type as push pull
      GPIOA->AFR[1] &= ~(0xFF << 4);//clearing the bits in the AFR register for pins 9 & 10
      GPIOA->AFR[1] |= (  ( 7 << (9-8)*4 ) | ( 7 << (10-8)*4) );

      /* In the above line, 2nd expression : 7 on LHS refers to the AF7 value in decimal,
        refer to rm-stm32f29, pg no. 289.
        (10 - 8) => 10th pin, - 8 because the AFRH starts from 8th bit, and *4 because we want to
      cover the bits
      */
      // The same expression could be done for clearing as well, since it was continuous 1s,
      // and the pins were next to each other i simply chose to give FF to clear out 8 bits
      countinously

 //clk is 16MHz, crystal oscillator (HIS)
 //USARTDIV=fclk/(16*Baud)
 //For baud rate 9600, mantissa = 104, fraction = .166 * 16 = 3
  USART1->BRR = (104<<4) | 3;

      USART1->CR1=0;
      USART1->CR1 |=  USART_CR1_TE | USART_CR1_RE;//Setting TE and RE bit
      USART1->CR1 |=   USART_CR1_UE; //(1 << 13)Setting the UE bit
      USART1->CR1 &= ~(USART_CR1_M);//~(1<<12);//Setting(0) M bit as 8
      USART1->CR1 &= ~(USART_CR1_OVER8);// ~(1<<15);Setting OVER8 as oversampling by 16.
      USART1->CR2  &=~(3<<(12));//setting stop bits as 00

}
void send_char(char ch)
{
      while(!(USART1->SR & USART_SR_TXE));//Waits until the data is tranferred from TDR to shift
      register
      USART1->DR=ch;//Transmits the data to data register.
}
void send_string(char *str)
{
        while(*str)
        {
        send_char(*str++);
        }
}

unsigned char getc(void)
{
        while(!(USART1->SR & USART_SR_RXNE));//Waits until a data is received, RXNE means Rx not
        Empty
        return USART1->DR;
}

int main()
{
        usart_init();
        delay();
        unsigned char c;


        while(1)
        {
                send_string("Hello stm32\n\r");
                //This is not working, but we comment the receive part, we can see this message
                delay();

               /* NOTE : Set the baud rate in Tera Term before connecting */

                   c=getc();//Type a character and it will be displayed
                send_string("\r\nInput is : ");
                send_char(c);


        }


}
```