Mircoprocessor Lab9 Report

2.Problem2 The ADC and UART communication

(1) What is UART?

Universal Asynchronous Receiver/Transmitter which is used to transmit the data through the serial port, it can be customised and connected to the pin in STM32 port.

(2) How to set up the UART ?

**void** USART1\_Init(**void**)

{

//f CK can be f LSE , f HSI , f PCLK , f SYS .,we can just use the clock from STM32, which is 4MHz

RCC->APB2ENR |= RCC\_APB2ENR\_USART1EN;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*UART CR1 setting\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CR1 may reference to p1346 of the manual

CR1 clear the bits of M(Data length/8bits is default) PS() PCE(Parity check) TE RE, then set the bits of TE RE where TE enables the function of transmittion and

RE enables the function of reception

In short, M bits=0-->Data frame to be 8 bits (this is suitable for ASCII Character transmittion)

PS bit=0-->Evan parity (this is the deault status for the UART transmittion in the picocom terminal ,or maybe in the most serial terminal?)

PCE bit=0-->Parity checking disable (this is the deault status for the UART transmittion in the picocom terminal ,or maybe in the most serial terminal?)

TE bit=1-->Transmitting enable

RE but=1--> Receiving enable

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

MODIFY\_REG(USART1->CR1, USART\_CR1\_M | USART\_CR1\_PS | USART\_CR1\_PCE | USART\_CR1\_TE | USART\_CR1\_RE |

USART\_CR1\_OVER8, USART\_CR1\_TE | USART\_CR1\_RE);

// CR2 for how much bit indicating the stop, now 1 bit

MODIFY\_REG(USART1->CR2, USART\_CR2\_STOP, 0x0); //0x0 for 1-bit stop

// CR3 clear the bits of RTSE , CTSE and ONEBIT, these are used for RS232, different from our LAB9

MODIFY\_REG(USART1->CR3, (USART\_CR3\_RTSE | USART\_CR3\_CTSE | USART\_CR3\_ONEBIT), 0x0);

/\*uint16\_t brr15\_4 = USART1->BRR & 0b1111111111110000;

brr15\_4 >>= 4;

uint16\_t brr2\_0 = USART1->BRR & 0b111;

brr2\_0 <<= 1;

uint16\_t baud\_x = brr15\_4 | brr2\_0;\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Baud rate setting,\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

oversampling by16 (since over8 is cleared)

USARTDIV is how fast the communication port in computer wants to transmit and receive (they should be the same value)

default terminal setting is 9600, then we set the baud rate = fCK/USARTDIV , for default fCK = fSYS --> 4M / 9.6K

or check manual p 1319 for 72MHz--> BRR with OVER16 USARTDIV = 9600-->1D4C so for 4MHz is 1D4C/18 about 416(DEC) which = 4MHz/9.6K

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

MODIFY\_REG(USART1->BRR, 0xFFFF /\*clear all and reset\*/, 4000000/9600);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*asynchronous mode setting\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In asynchronous mode, the following bits must be kept cleared:

- LINEN and CLKEN bits in the USART\_CR2 register,

- SCEN, HDSEL and IREN bits in the USART\_CR3 register.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

USART1->CR2 &= ~(USART\_CR2\_LINEN | USART\_CR2\_CLKEN);

USART1->CR3 &= ~(USART\_CR3\_SCEN | USART\_CR3\_HDSEL | USART\_CR3\_IREN);

// Enable UART

USART1->CR1 |= (USART\_CR1\_UE);

}