<<IERG3810>>

<<Microcontroller and Embedded Systems Laboratory>>

Report on Experiment <<2>>

<<Universal Synchronous/Asynchronous Receiver/Transmitter (USART) >>

Group: 19

Member: Chan Kai Yin (CUID: 1155124983)

Ip Tsun Yu (CUID: 1155144668)

Submission Date: 16-02-2022

Disclaimer

I declare that the assignment here submitted is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/

Jesse	Chan Kai Yin	16-02-2022
Signature	Name	Date
Derek	Ip Tsun Yu	16-02-2022
Signature	Name	Date

I. OBJECTIVES

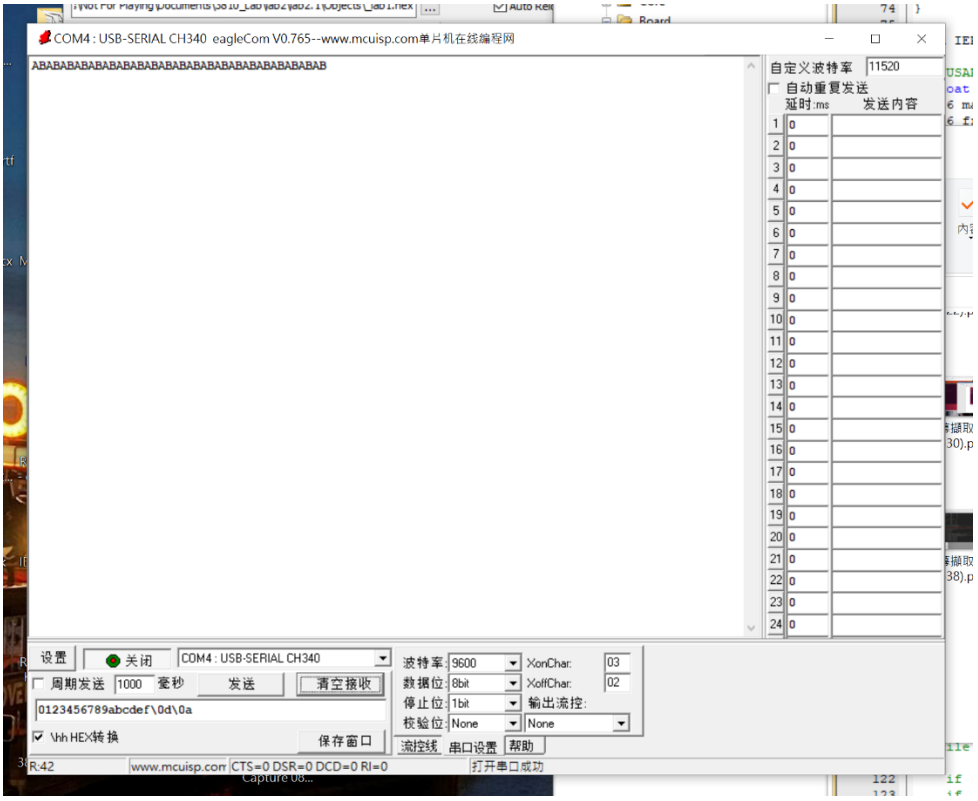
To study the clock tree of Cortex-M3.

To study the settings of USART transmitting.

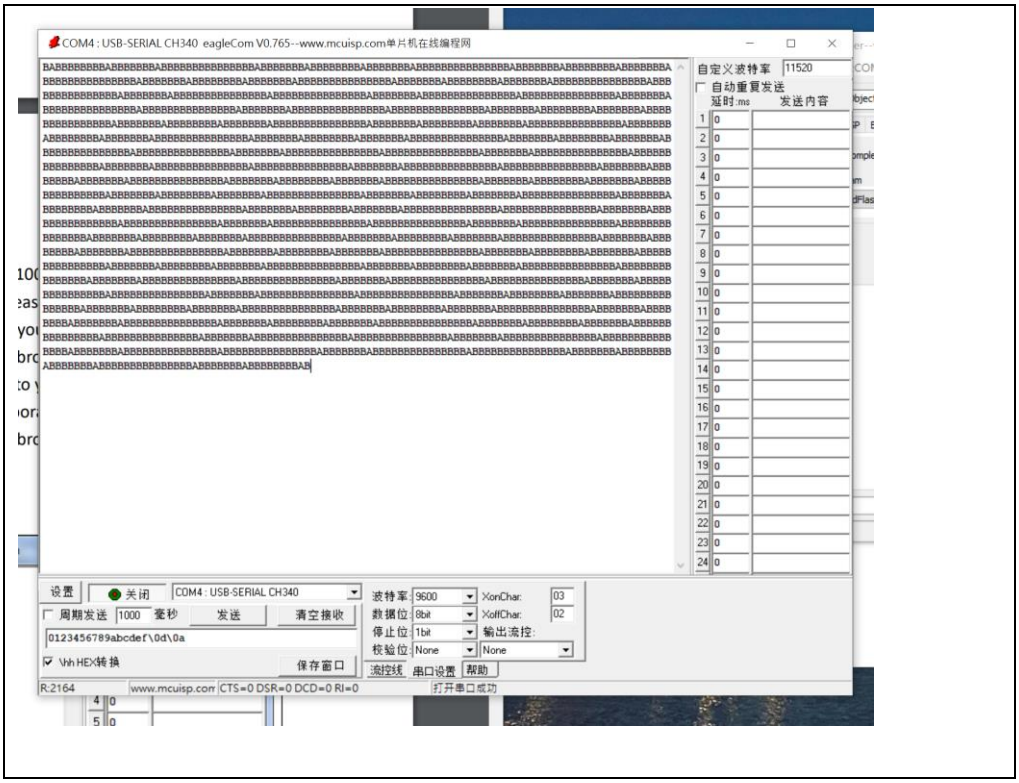
II. DATA ANALYSIS

Experiment 2.1

The alphabets AB shown on the software:

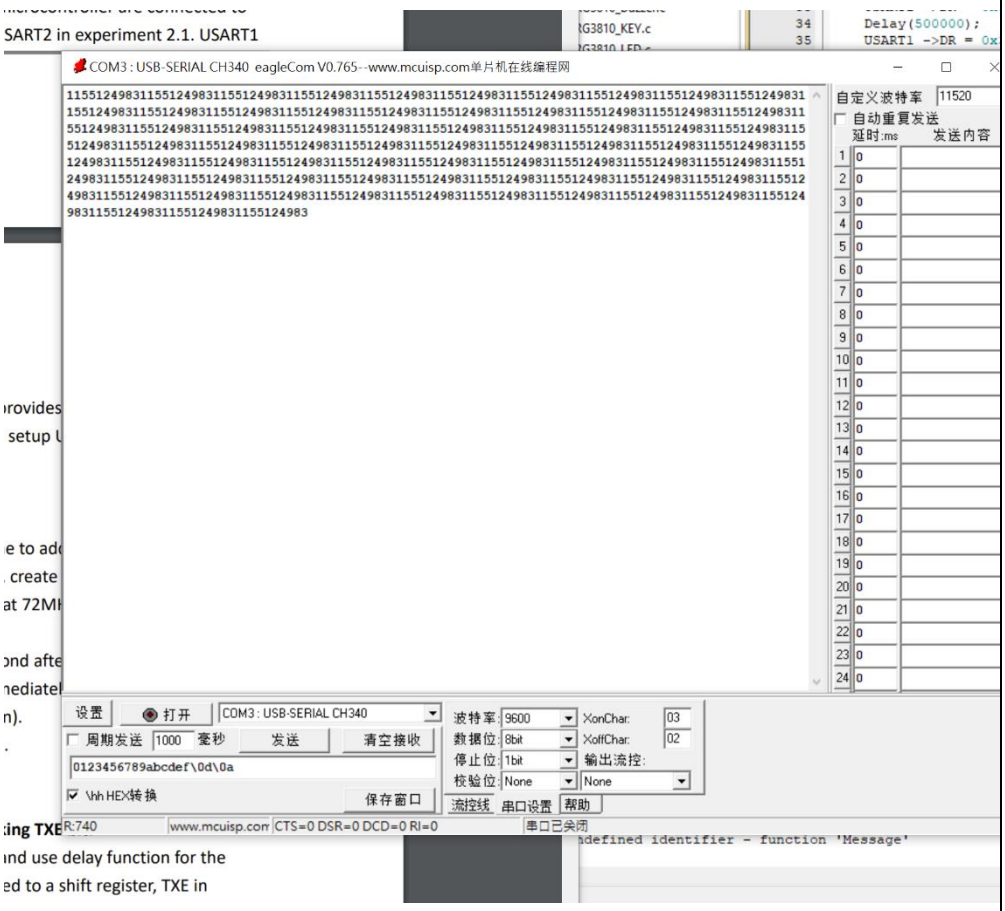


After modifying the figure of ‘Delay(50000)’ to ‘Delay(100)’:


COM4 : USB-SERIAL CH340 eagleCom V0.765---www.mcuisp.com单片机在线编程网
自定义波特率
11520
自动重复发送
延时:ms
发送内容
设置
关闭
COM4 : USB-SERIAL CH340
周期发送
1000
毫秒
发送
清空接收
0123456789abcdef\0d\0a
\hh HEX转换
保存窗口
波特率
9600
数据位
8bit
停止位
1bit
校验位
None
XonChar:
03
XoffChar:
02
输出流控:
None
流控线
串口设置
帮助
R:2164
www.mcuisp.com
CTS=0 DSR=0 DCD=0 RI=0
打开串口成功

Experiment 2.2

WithSID1155124983:




```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void Delay(u32);

void Delay(u32 count)
{
	u32 i;
```

```
	for(i=0;i<count;i++);
}

void IERG3810_clock_tree_init(void){
	u8 PLL=7;
	unsigned char temp=0;
	RCC->CFGR &= 0xF8FF0000;

	RCC->CR &= 0XFEF6FFFF;
	RCC->CR |= 0x00010000;
	while(!(RCC->CR>>17));
	RCC->CFGR=0x00000400;
	RCC->CFGR|=PLL<<18;
	RCC->CFGR|=1<<16;

	FLASH->ACR|=0x32; // 72
	RCC->CR|=0x01000000;
	while(!(RCC->CR>>25));
	RCC->CFGR|=0x00000002;

	while(temp!=0x02)
	{
		temp=RCC->CFGR>>2;
		temp&=0x03;
	}
}

void IERG3810_USART2_init(u32 pclk1, u32 bound)
{
	//USART2
	float temp;
	u16 mantissa;
	u16 fraction;
	temp= (float) (pclk1*1000000)/(bound*16);
	mantissa = temp;
	fraction = (temp - mantissa)*16;
	mantissa <<= 4;
```

```
	mantissa += fraction;
	RCC -> APB2ENR |= 1<<2;
	RCC -> APB1ENR |= 1<<17;
	GPIOA -> CRL &= 0XFFFF00FF;
	GPIOA -> CRL |= 0X00008B00;
	RCC -> APB1RSTR |= 1<<17;
	RCC -> APB1RSTR &= ~(1<<17);
	USART2 -> BRR = mantissa;
	USART2 -> CR1 |= 0X2008;
}

void IERG3810_USART1_init(u32 pclk2, u32 bound)
{
	float temp;
	u16 mantissa;
	u16 fraction;
	temp = (float) (pclk2*1000000)/(bound*16);
	mantissa = temp;
	fraction = (temp - mantissa)*16;
	mantissa<<= 4;
	mantissa += fraction;
	RCC->APB2ENR |= 1<<2;
	RCC->APB2ENR |= 1<<14;
	GPIOA->CRH &= 0XFFFFF00F;
	GPIOA->CRH |= 0X000008B0;
	RCC->APB2RSTR |= 1<<14;
	RCC->APB2RSTR &= ~(1<<14);
	USART1->BRR = mantissa;
	USART1->CR1 |=0X2008;
}

int main(void)
{
	IERG3810_clock_tree_init();
	IERG3810_USART2_init(36, 9600);
	IERG3810_USART1_init(72, 9600);
```

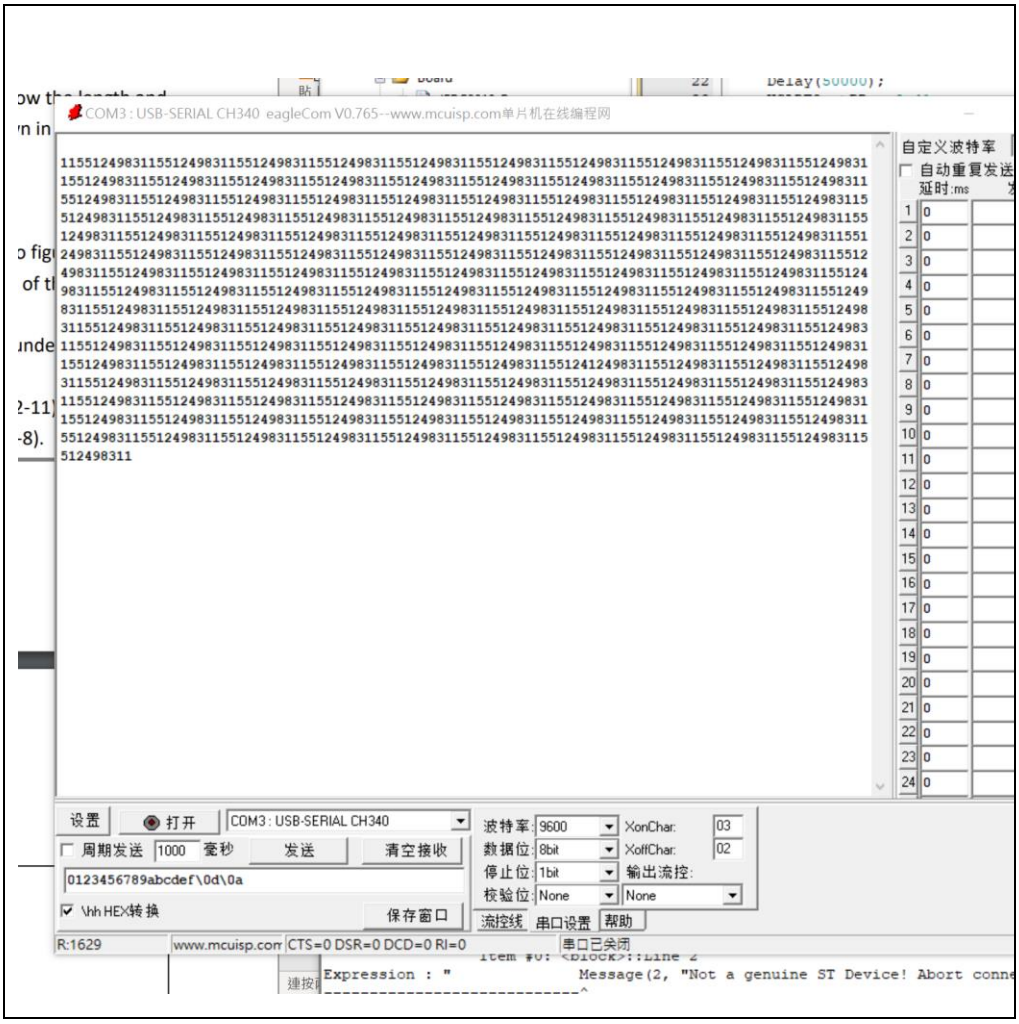
```
	while(1)
	{
		//USART2 ->DR = 0x41; // A
		Delay(50000);
		//USART2 ->DR = 0x42; // B

		// 1155124983
		USART1 -> DR = 0x31;
		Delay(50000);
		USART1 -> DR = 0x31;
		Delay(50000);
		USART1 -> DR = 0x35;
		Delay(50000);
		USART1 -> DR = 0x35;
		Delay(50000);
		USART1 -> DR = 0x31;
		Delay(50000);
		USART1 -> DR = 0x32;
		Delay(50000);
		USART1 -> DR = 0x34;
		Delay(50000);
		USART1 -> DR = 0x39;
		Delay(50000);
		USART1 -> DR = 0x38;
		Delay(50000);
		USART1 -> DR = 0x33;
		Delay(50000);

		Delay(1000000);

	}
}
```

Experiment 2.3



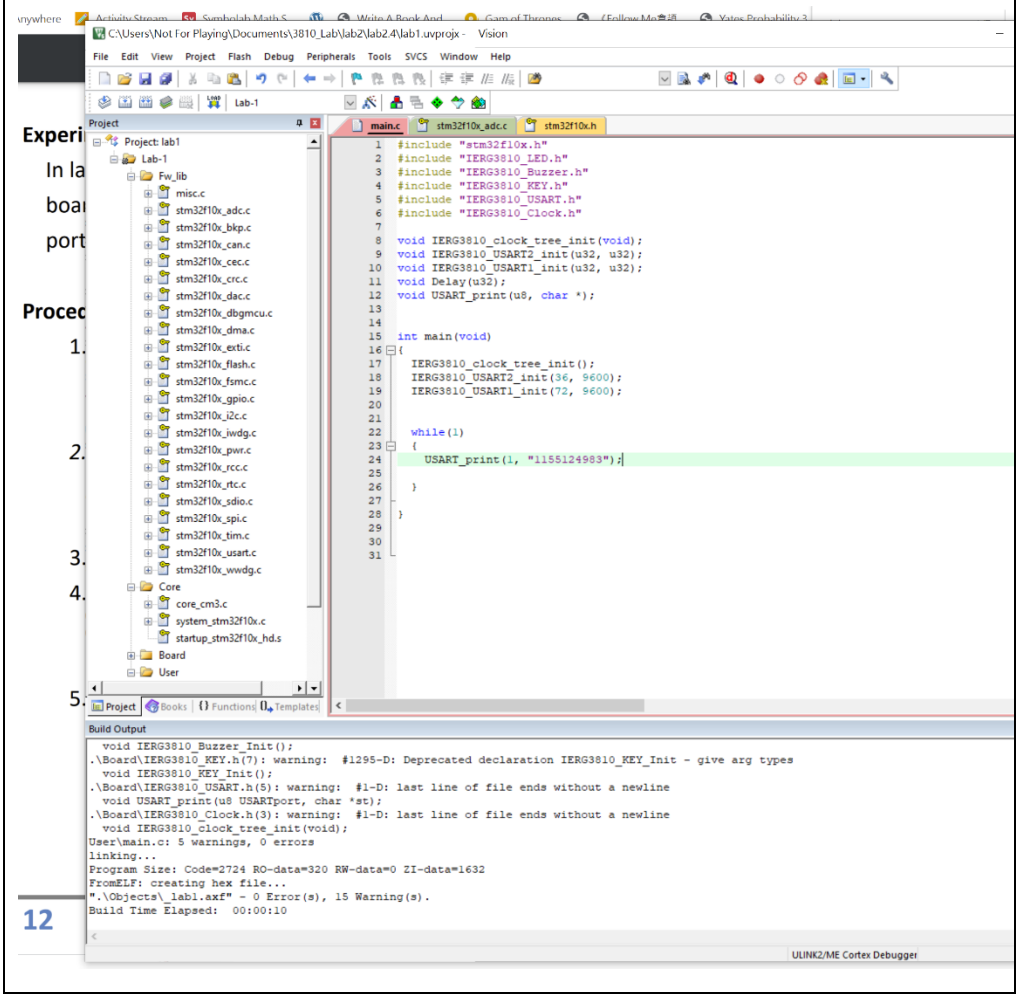
Replace the Delay(50000) command in USART_print()

```
void USART_print(u8 USARTport, char *st)
{
	u8 i=0;
	while (st[i] != 0x00)
	{
		if (USARTport == 1) USART1->DR = st[i];
		if (USARTport == 2) USART2->DR = st[i];
		while (((USART1 -> SR) & (0x00000080))==0);
		while (((USART2 -> SR) & (0x00000080))==0);
```

```
            //Delay(50000);
            while (!(USART1->SR << 7));
            if (i == 255) break;
            i++;
        }
}
```

Experiment 2.4

Only keep ‘main()’ function in ‘main.c’ file. Compile the program and download it to the board without error.



III. DISCUSSION

Exp 2.1

<Step 8 : Describe what has happened and the reason of an error of the serial communication (check eagleCom's screen.>

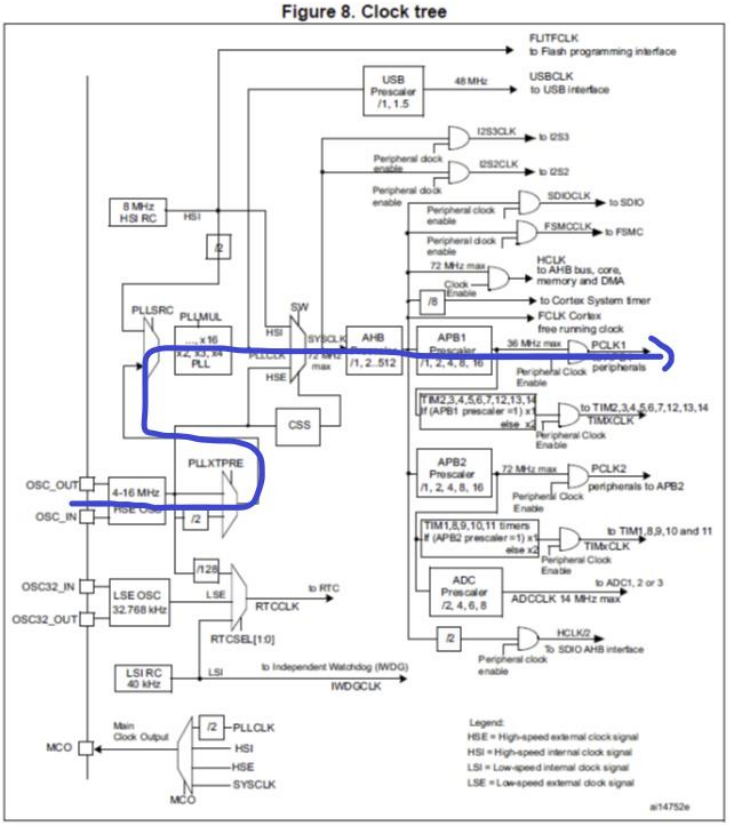
The speed of popping is faster than before. Also, A is popping up once combining with multiple B and repeated. It should have the same effect as previous which is repeating "ABAB". It may due to data corruption because of the pervious character has not been sent and the new character already launch.

< Step 9: Describe the setting and procedures of the subroutine 'IERG3810_clock_tree_init()'>

It clears first 16 bits and conf of clock output on CFGR, setting PLL off, Clock detector OFF, HSE oscillator ON. Check for internal 8 MHz RC oscillator ready, HCLK divided by 2, PLL input clock x 9, HSE clock divided by 2, Two wait states, if 48 MHz < SYSCLK <= 72 MHz. PLL ON, check PLL clock ready flag, PLL selected as system clock. While loop until the PLL used as system clock, set the temp = System clock switch status and keep last two bits.

<The signal path>

Figure 8. Clock tree



< Step 10: Describe the setting and procedures of the subroutine 'IERG3810_USART2_init()'>

Baud is calculated by the above formula. It further divided into integer and fraction and let them as mantissa and fraction. GPIOA and USART2 are enabled. Setting USART2 as alternative functions with PA2 and PA3. Set the mantissa of USARTDIV and Transmitter enable.

Exp 2.3

<Step 4 : Try to understand the function of USART_print().>

The function takes the port number of USART and the target printed char. Then it iterates the string char by char. It sends to the specify port number then wait for data is transferred to the shift register. It continues to send the next char in the string until the last char is send.

IV. SUMMARY

We learn the setting of clock tree and USART transmitting of Cortex-M3.

V. DIVISION OF WORK

<Lab work: Jesse & Derek, Report writing: Jesse>

VI. REFERENCES