

```

//*****
// Name: Benjamin Cesero
// Date Due: 11/28/22
// Course: ELEC3371
// Description:
// This program enables USART2 while running at 36mhz. Sending 1 character in
// the terminal will respond with that character back.
//*****
// Global Varibales & Includes
volatile int rcvrd;
// Function Declarations
void InitializeUSART1();
void InitializeUSART2();

void main()
{
    InitializeUSART1(); // Call InititalizeUSART1 function
    InitializeUSART2(); // Call InititalizeUSART2 function
    for(;;)
    {
        while (!((USART2_SR & (1<<5)) == 0x20)){}
        rcvrd = USART2_DR;
        while (!(USART2_SR & (1<<7)) == 0x80) {}
        USART2_DR = rcvrd;
    }
}

void InitializeUSART1() // Sub function which initializes the registers to
enable USART1
{
    RCC_APB2ENR |= 1; // Enable clock for Alt. Function. USART1
    uses AF for PA9/PA10
    AFIO_MAPR=0X0F000000; // Do not mask PA9 and PA10 (because we are
using for USART)
    RCC_APB2ENR |= 1<<2; // Enable clock for GPIOA
    GPIOA_CRH &= ~(0xFF << 4); // Clear PA9, PA10
    GPIOA_CRH |= (0x0B << 4); // USART1 Tx (PA9) output push-pull
    GPIOA_CRH |= (0x04 << 8); // USART1 Rx (PA10) input floating
    RCC_APB2ENR |= 1<<14; // enable clock for USART1
    USART1_BRR=0X00000506; // Set baud rate to 56000
    USART1_CR1 &= ~(1<<12); // Force 8 data bits. M bit is set to 0.
    USART1_CR2 &= ~(3<<12); // Force 1 stop bit
    USART1_CR3 &= ~(3<<8); // Force no flow control and no DMA for USART1
    USART1_CR1 &= ~(3<<9); // Force no parity and no parity control
    USART1_CR1 |= 3<<2; // RX, TX enable
    USART1_CR1 |= 1<<13; // USART1 enable. This is done after
configuration is complete
    Delay_ms(100); // Wait for USART to complete configuration and
enable. This is // not always necessary, but good practice.
}

void InitializeUSART2()
{
    USART2_CR1.UE = 0; // Disable USART to configure
    RCC_APB1RSTR.USART2RST = 0; // Don't hold USART2 in reset
    RCC_APB2ENR.AFIOEN = 1; // Enable clock for Alt. Function. USART2 uses
Alternate pins PD5/PD6
}

```

```

RCC_APB2ENR.IOPDEN = 1;    // Enable clock for GPIOD
GPIOD_CRL = 0;
GPIOD_CRL |= 0xB000000; //(0xB<<20); // Set PD5 output push pull
GPIOD_CRL |= 0x40000000; //(0x4<<24); //Set PD6 input floating
AFIO_MAPR.USART2_REMAP = 1; // remap to TX/PD5, RX/PD6
RCC_APB1ENR.USART2EN = 1; // enable USART2 on the APB1ENR
USART2_BRR=0x000000EA6;    // set baud rate to 9600bps with 36mhz clk
USART2_CR1 &= ~(1<<12);    // Force 8 data bits. M bit is set to 0.
USART2_CR2 &= ~(3<<12);    // Force 1 stop bit
USART2_CR3 &= ~(3<<8);     // Force no flow control and no DMA for USART1
USART2_CR1 &= ~(3<<9);     // Force no parity and no parity contro
USART2_CR1 |= 3<<2;        // RX, TX enable
USART2_CR1.UE = 1;        // Re-enable USART
Delay_ms(100);            // Wait for USART to complete config
}

```