--WITH INTERRUPTS

#include "stm32f4xx.h"

// Function to initialize USART2 for UART communication

void USART2\_Init() {

// Enable USART2 and GPIOA clock

RCC->APB1ENR |= RCC\_APB1ENR\_USART2EN;

RCC->AHB1ENR |= RCC\_AHB1ENR\_GPIOAEN;

// Configure PA2 (TX) and PA3 (RX) as alternate function

GPIOA->AFR[0] |= (7 << (4 \* 2)) | (7 << (4 \* 3));

GPIOA->MODER |= GPIO\_MODER\_MODER2\_1 | GPIO\_MODER\_MODER3\_1;

// Configure USART2

USART2->BRR = 0x0683; // 115200 baud @ 16MHz APB1 clock

// Enable USART2 receive and transmit interrupts

USART2->CR1 = USART\_CR1\_TE | USART\_CR1\_RE | USART\_CR1\_UE | USART\_CR1\_RXNEIE | USART\_CR1\_TCIE;

// Enable USART2 global interrupt in NVIC

NVIC\_EnableIRQ(USART2\_IRQn);

NVIC\_SetPriority(USART2\_IRQn, 0); // Set the interrupt priority

}

// Function to send a character over USART2

void USART2\_SendChar(char ch) {

while (!(USART2->SR & USART\_SR\_TXE));

USART2->DR = ch;

}

// USART2 interrupt handler

void USART2\_IRQHandler(void) {

if (USART2->SR & USART\_SR\_RXNE) {

char received\_char = USART2->DR; // Read the received character

// Echo the received character back

USART2\_SendChar(received\_char);

}

}

int main(void) {

// Initialize system

SystemInit();

// Initialize USART2

USART2\_Init();

while (1) {

// You can perform other tasks here

// For example, periodically send data over USART2

USART2\_SendChar('A'); // Send a character

}

}

--WITHOUT USING INTERRUPTS

#include "stm32f4xx.h"

// Function to initialize UART2

void UART2\_Init() {

// Enable UART2 and GPIOA clock

//RCC- RESET AND CLOCK CONTROL

RCC->APB1ENR |= RCC\_APB1ENR\_USART2EN; bit 17

RCC->AHB1ENR |= RCC\_AHB1ENR\_GPIOAEN; bit 0

// Configure PA2 (TX) and PA3 (RX) as alternate function

//AFRL

GPIOA->AFR[0] |= (7 << (4 \* 2)) | (7 << (4 \* 3));

GPIOA->MODER |= GPIO\_MODER\_MODER2\_1 | GPIO\_MODER\_MODER3\_1;

// Configure UART2

USART2->BRR = 0x683; // 9600 baud @ 16MHz APB1 clock

USART2->CR1 = USART\_CR1\_TE | USART\_CR1\_RE | USART\_CR1\_UE;

}

// Function to send a character over UART2

void UART2\_SendChar(char ch) {

while (!(USART2->SR & USART\_SR\_TXE));

USART2->DR = ch;

}

// Function to receive a character from UART2

char UART2\_ReceiveChar() {

while (!(USART2->SR & USART\_SR\_RXNE));

return USART2->DR;

}

int main(void) {

// Initialize system

SystemInit();

// Initialize UART2

UART2\_Init();

char rxData;

while (1) {

// Receive data from UART and echo it back

rxData = UART2\_ReceiveChar();

UART2\_SendChar(rxData);

}

}