```c
/*
 * USART3_loopback.c
 *
 * Created: 2/27/2022 3:24:01 PM
 * Author : jason
 */

#include <avr/io.h>
#define F_CPU 4000000
#include <util/delay.h>
#include <avr/interrupt.h>

#define USART3_BAUD_RATE(BAUD_RATE) ((float)(F_CPU * 64 / (16 *(float)BAUD_RATE)) +
  0.5)  //Calculation of baud rate from data sheet


char c = 'A';


ISR(USART3_RXC_vect){
    //Poll until there is data to be received
    while(!(USART3.STATUS & USART_RXCIF_bm)){}

    c = USART3_RXDATAL;
    if(c >= 'A' && c <= 'Z'){
        //Poll until the  transmit buffer register are empty
        //when they contain data that has not been moved to
        //transmit shift register
        while (!(USART3.STATUS & USART_DREIF_bm)){}
        //Convert to lowercase letter
        USART3.TXDATAL = c + 0x20;

    }
    else if(c >= 'a' && c < 'z'){
        //Poll until the  transmit buffer register are empty
        //when they contain data that has not been moved to
        //transmit shift register
        while (!(USART3.STATUS & USART_DREIF_bm)){}
        //Convert to uppercase except for 'z' since next letter would be 'A'
        USART3.TXDATAL = c - 0x1F;
    }
    else if (c == 'z'){
        //Poll until the  transmit buffer register are empty
        //when they contain data that has not been moved to
        //transmit shift register
        while (!(USART3.STATUS & USART_DREIF_bm)){}
        USART3.TXDATAL = 'A';  //Convert to A if letter is 'z'
    }

}

int main(void)
```

```
{
    PORTB.DIR |= PIN0_bm;  //Set PB0 as output (TX pin) and PB1 in input pin (RX pin)
    USART3.BAUD = (uint16_t)USART3_BAUD_RATE(9600);   //Taken from data sheet to
      calculate baud rate
    USART3.CTRLB |= USART_TXEN_bm | USART_RXEN_bm;   //Enable USART transmitter
    USART3.CTRLA |= USART_LBME_bm | USART_RXCIE_bm;  //Enable internal connection
      between the TXD pin and the USART receiver
                                                     //and the RX input of the USART
                    receiver is disconnected. Also enable RXCIE interrupt

    //Poll until the  transmit buffer register are empty
    //when they contain data that has not been moved to
    //transmit shift register
    while (!(USART3.STATUS & USART_DREIF_bm)){}
    USART3_TXDATAL = c;

    sei();   //Enable global interrupt


    while (1)
    {
        //Do nothing
        asm volatile("nop");
    }
}
```