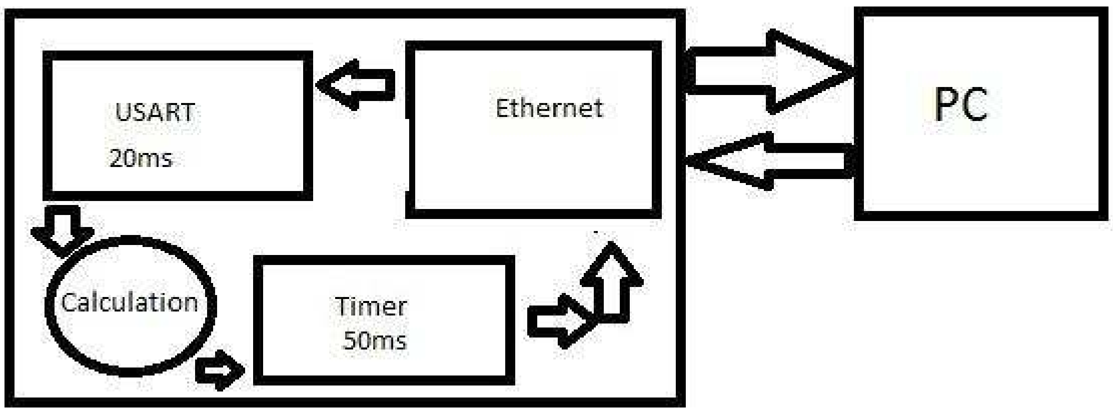
Hello
Here is my circuit diagram



Here is my program

```
volatile  unsigned char flagSerial;
volatile unsigned char flagEthernet;
volatile unsigned char bitpos1;
volatile unsigned char endbitpos1;
volatile unsigned char stat1;
volatile unsigned char SendSerial;
volatile unsigned char temp;
volatile unsigned char bufferu1[200];
volatile unsigned char buffer1[200];
unsigned char pc[100];

int32_t socket;
uint8_t *sendbuf;
uint8_t buf[8];



int main (void)
{
        netInitialize ();
        USART();
        NVICS();
        Tim();
        flagEthernet=0;
        flagSerial=0;
        bitpos1=0;
        SendSerial=0;
        stat1=0;

        socket = netTCP_GetSocket (tcp_cb_func);
        if (socket >= 0) {
         netTCP_Listen (socket, 3000);
        }


                while (1)
                {
                        if(flagEthernet==1){
                        //some calculation
                        send();flagEthernet=0;}
                        if(flagSerial==1){
                        //some calculation
                        send();
                        flagSerial=0;}
                }
}

uint32_t tcp_cb_func (int32_t socket, netTCP_Event event,const NET_ADDR *addr, const uint8_t *buf,
uint32_t len) {
   switch (event) {

    case netTCP_EventConnect:return (1);
    case netTCP_EventData:flagEthernet=1;memcpy(pc, buf, len);break;
                }
  return (0);
}

void NVICS(void){
    NVIC_InitTypeDef nvicStructure;
    nvicStructure.NVIC_IRQChannel = TIM2_IRQn;
    nvicStructure.NVIC_IRQChannelPreemptionPriority = 0;
    nvicStructure.NVIC_IRQChannelSubPriority = 1;
    nvicStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&nvicStructure);
}


void Tim(void){
TIM_TimeBaseInitTypeDef TimeStruct;
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2,ENABLE);
TimeStruct.TIM_Prescaler=7200;
TimeStruct.TIM_Period=500;
TimeStruct.TIM_ClockDivision=TIM_CKD_DIV1;
TimeStruct.TIM_CounterMode=     TIM_CounterMode_Up ;
TimeStruct.TIM_RepetitionCounter        =0;
TIM_TimeBaseInit(TIM2, &TimeStruct);
TIM_Cmd(TIM2, ENABLE);
TIM_ITConfig(TIM2,TIM_IT_Update,ENABLE);
}


void TIM2_IRQHandler(){
           if (TIM_GetITStatus(TIM2, TIM_IT_Update) != RESET)
    {
                sendPC();
                TIM_ClearITPendingBit(TIM2, TIM_IT_Update);
                }
        }

void sendPC(void){
        unsigned char i;

                        if( netTCP_GetState(socket)==netTCP_StateESTABLISHED)
                        {
                        if (netTCP_SendReady (socket))
                        {

sprintf(msg,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCC

                                        for(i=0;i<150;i++)
                                        {
                                                if(msg[i]=='\0')
                                                        {
                                                        break;
                                                        }
                                        }

                if (netTCP_SendReady (socket)) {
```

```
                sendbuf =  netTCP_GetBuffer (i);
                memcpy (sendbuf, msg, i);
                netTCP_Send(socket, sendbuf,i);

                        }
                }
        }
}

void send(void){
        int i;
        for(i=0;i<150;i++){
                msg[i]='\0';
        }
        sprintf(msg,"AAAAAAAAAAAAAABBBBBBBBBBBBBBBDDDDDDDDDDDDDDDDDEEEEEEEEEEEEEEEFFFFFFFFFFF");
        count1=0;
        SendSerial=1;
        USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
}

void USART(void){
        USART_InitTypeDef USAR;
        GPIO_InitTypeDef GPIOStruc;


        RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB |RCC_APB2Periph_AFIO,ENABLE);
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1,ENABLE);
        GPIO_PinRemapConfig(GPIO_Remap_USART1,ENABLE);

        GPIOStruc.GPIO_Mode=GPIO_Mode_AF_PP ;
        GPIOStruc.GPIO_Speed=GPIO_Speed_50MHz;
        GPIOStruc.GPIO_Pin=GPIO_Pin_6;
        GPIO_Init(GPIOB,&GPIOStruc);

        GPIOStruc.GPIO_Mode=GPIO_Mode_IN_FLOATING ;
        GPIOStruc.GPIO_Pin=GPIO_Pin_7;
        GPIO_Init(GPIOB,&GPIOStruc);

        USAR.USART_BaudRate=115200;
        USAR.USART_StopBits=USART_StopBits_1;
        USAR.USART_WordLength=USART_WordLength_8b;
        USAR.USART_Parity=USART_Parity_No ;
        USAR.USART_HardwareFlowControl=USART_HardwareFlowControl_None;
        USAR.USART_Mode=USART_Mode_Rx | USART_Mode_Tx;
        USART_Init(USART1,&USAR);
        USART_ITConfig(USART1,USART_IT_RXNE,ENABLE);
        NVIC_EnableIRQ(USART1_IRQn);
        USART_Cmd(USART1,ENABLE);
}


void USART1_IRQHandler(void){

        if (USART_GetITStatus(USART1, USART_IT_TXE) != RESET)
  {

                                if(SendSerial==1 && count1<=150){
                                if(msg[count1]=='\0' ){USART_ITConfig(USART1, USART_IT_TXE,
DISABLE);SendSerial=0;return;}
                                USART_SendData(USART1,msg[count1]);
                                count1=count1+1;
                                }
                                if(SendSerial==0 || count1>=150){USART_ITConfig(USART1, USART_IT_TXE,
DISABLE);SendSerial=0;}
}
```

```
if (USART_GetITStatus(USART1, USART_IT_RXNE) != RESET)
{
        temp=USART_ReceiveData(USART1);
        if(SendSerial==1){return;}
        if(temp==65){
        stat1=1;
        bitpos1=0;
        }
        else if(temp!=66 && stat1==1 && temp>47 && temp<58){
        bufferu1[bitpos1]=temp;
        bitpos1+=1;
        }
        else if(temp==66 && stat1==1)
        {
        stat1=0;
        copy();
        endbitpos1=bitpos1;
        flagSerial=1;
        }else if(stat1==1 && bitpos1>150){
                bitpos1=0;
                stat1=0;
        }else if(bitpos1>150){
                bitpos1=0;
                stat1=0;
        }
}

}

void copy(void){
        char i;
        for(i=0;i<100;i++){buffer1[i]=bufferu1[i];}
}
```

Above program works fine (receive data from USART and it print data to Ethernet each 50ms very well). However after **some** transmission between PC to Ethernet it hangs and only I see "Hard Fault" on network bus.I think
1)when I'm in receiving mode(from PC) and Timer try to send data to PC it hangs , how can prevent from sending while we are in receiving mode?(because when I call netTCP_Send(socket, sendbuf,i); it sends all data without any control)(But I think it is be possible to send and receive simultaneously such as send and receive with USART interrupts)

2)After sending a string from pc to my device I've seen it works fine if I change this function

```
void send(void){
        int i;
        for(i=0;i<150;i++){
                msg[i]='\0';
        }
        sprintf(msg,"AAAAAAAAAAAAAABBBBBBBBBBBBBBDDDDDDDDDDDDDDDDDEEEEEEEEEEEEEEFFFFFFFFFF");

        return;
        count1=0;
        SendSerial=1;
        USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
}
```