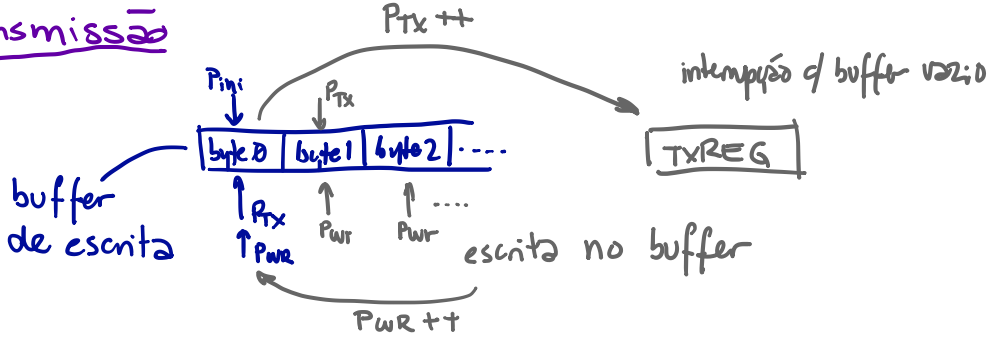
Comunicação serial por interrupção

Transmissão

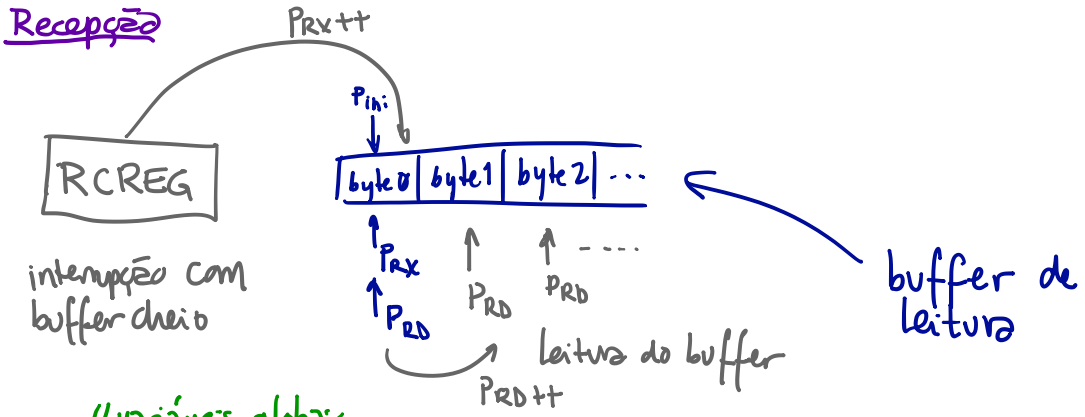



```
//variáveis globais
volatile char wr_buffer[20]; //buffer c/ 20 chars
volatile char *pini; //ponteiro para inicio do buffer
volatile char *ptx; //ponteiro de transmissão
volatile char *pwr; //ponteiro de escrita no buffer

//rotina de tratamento de interrupções
void interrupt isr(){
  //interrupção de transmissão serial
  if(TXIE && TXIF){  //se buffer de Tx vazio
     if(ptx<pwr){  //se tem char p/ transmitir
        TXREG=*ptx++; //escreve no buffer e incr. ponteiro
     }//fim-transferência para buffer
     if(ptx==pwr){ //se ponteiros forem iguais
        ptx=pini; //retorna ponteiro de tx p/ inicio do buffer
        pwr=pini; //retorna ponteiro de escrita p/ inicio do buffer
     }//fim-ponteiros iguais
  }//fim-interrupção de Tx
}//fim-rotina de tratamento de interrupções
```

```
//inicialização no main
pini = wr_buffer; // ponteiro p/ inicio do buffer
ptx= pini; // ponteiro de tx aponta p/ inicio do buffer
pwr = ptx; // ponteiro de escrita coincide c/ ponteiro de transmissão
TXIE=1; //habilita interrupção de transmissão serial

//no loop principal
*pwr++ = 'A'; // escreve char no buffer
```




```
//variáveis globais
volatile char rd_buffer[20]; //buffer de leitura c/ 20 char
volatile char *pini; // ponteiro p/ inicio do buffer
volatile char *prx; //ponteiro de recepção
volatile char * prd: // ponteiro de leitura

//rotina de tratamento de interrupções
void interrupt isr(){
    //interrupção de recepção serial
    if(RCIE && RCIF) { // se buffer de recepção cheio
        *prx++ = RXREG;// transfere p/ buffer de leitura
    } // fim - buffer RX cheio e incrementa ponteiro
}// fim - rotina de tratamento de interrupção
```

```
// variáveis locais no main
char cread; // variável ou vetor p/ receber caracteres lidos


//inicializações no main
pini = rd_buffer; // ponteiro para inicio do buffer
prx = pini; // ponteiro de recepção aponta para inicio do buffer
prd = prx; // ponteiro de leitura coincide com ponteiro de Rx
RCIE=1; //habilita interrupção de recepção serial


//no loop principal
if(prd < prx) { // se chegou caracter mas não foi lido
    cread = *prd++; // retira do buffer de leitura e incr. ponteiro
} //fim - transferência para buffer
if( prd == prx) { // se ponteiros forem iguais
    prd = pini; //retorna ponteiro de leitura para inicio do buffer
    prx = pini; // retorna ponteiro de recepção para inicio do buffer
} //fim ponteiros iguais
```

variável ou vetor p/ receber caracteres