Sender StateChart for XMODEM file-transfer protocol.
March 25, 2014
Real events are either
SER (serial character recieved),
KB_C (cancel command received from keyboard)
TM (timeout occurred)

Ordinarily...

TM_VL (Very Long timeout) gives 60 seconds
TM_SOH (normal timeout waiting for SOH) gives 10 seconds
TM_CHAR (inter-character timeout) gives 1 second
TM_2CHAR gives a period longer than the inter-character timeout of 1 sec.

CompleteSenderTopLevel

[Entry]
ctx.useCrc=true; ctx.prep1stBlk(); ctx.errCnt=0;
ctx.firstCrcBlk=true; ctx.KbCan = false; ctx.tm(TM_VL);
[Exit]

ctx.can8() should not be called immediately after the checksum for a block has been written to the medium, or some CAN characters may be purged.

onEvent(KB_C)
[!ctx.KbCan]
/ctx.KbCan = true;
ctx.tmRed(TM_VL - TM_2CHAR);

H — SERcancelable

[Entry]
[Exit]

onEvent(SER)
[(c==NAK || c=='C') & ctx.bytesRd]
/if (c==NAK) {ctx.useCrc=false;
ctx.cs1stBlk();
ctx.firstCrcBlk=false;}
ctx.sendBlkPrepNext();
ctx.tm(TM_VL);

onEvent(SER)
[(c==ACK) & ctx.bytesRd & !ctx.KbCan]
/ctx.sendBlkPrepNext();
ctx.errCnt=0; ctx.firstCrcBlk=false;
ctx.tm(TM_VL);

START
[Entry]
[Exit]

ACKNAK
[Entry]
[Exit]

onEvent(SER)
[(c==NAK || (c=='C' & ctx.firstCrcBlk)) &
(ctx.errCnt < errB) & !ctx.KbCan]
/ctx.resendBlk();
ctx.errCnt++; ctx.tm(TM_VL);

onEvent(SER)
[(c==ACK) & !ctx.bytesRd & !ctx.KbCan]
/ctx.sendByte(EOT);ctx.errCnt=0;
ctx.tm(TM_VL);ctx.firstCrcBlk=false;

onEvent(SER)
[c != CAN & !ctx.KbCan]
/ctx.tmPop();

EOT1
[Entry]
[Exit]

onEvent(SER)
[(c == NAK) || (c == 'C')) &
!ctx.bytesRd]
/if (c==NAK)
{ctx.firstCrcBlk=false;}
ctx.sendByte(EOT);
ctx.tm(TM_VL);

onEvent(SER)
[c==NAK & !ctx.KbCan]
/ctx.sendByte(EOT);
ctx.errCnt=0;
ctx.tm(TM_SOH);

EOTEOT
[Entry]
[Exit]

onEvent(SER)
[c==NAK & (ctx.errCnt
< errB) & !ctx.KbCan]
/ctx.sendByte(EOT);
ctx.errCnt++;

onEvent(SER)
[(c=='C' &
ctx.firstCrcBlk) &
(ctx.errCnt < errB) &
!ctx.KbCan]
/ctx.sendByte(EOT);
ctx.tm(TM_VL);
ctx.errCnt++;

onEvent(TM)
/ctx.result="Timeout
waiting for ACK of
2nd EOT";

onEvent(TM)
[!ctx.KbCan]
/ctx.tmPop();

onEvent(SER)
[c == CAN]
/ctx.tmPush(TM_CHAR)

onEvent(SER)
[c==ACK]
/ctx.result="1st EOT ACK'd";
/*should delay TM_2CHAR */
ctx.sendByte(EOT);

CAN
[Entry]
[Exit]

onEvent(KB_C)
/ctx.KbCan=true;

onEvent(SER)
[c==ACK]
/ctx.result="Done";

onEvent(SER)
[ctx.KbCan]
/ctx.can8();
ctx.result="KbCancelled";

onEvent(SER)
[c==CAN]
/ctx.clearCan();
ctx.result="RcvCancelled";

onEvent(TM)
/ctx.can8();
if (ctx.KbCan)
ctx.result="KbCancelled";
else
ctx.result="Timeout";

onEvent(KB_C)
/ctx.can8();
ctx.result="KbCancelled";

onEvent(SER)
[((c=='C' & ctx.firstCrcBlk) || c==NAK) & !ctx.KbCan]
/ctx.can8();
ctx.result="ExcessiveNAKsOrCs";