//---------------------------//  
//---------------------------//  
//  System total status      //  
//---------------------------//  
//---------------------------//  
  
// SEND : PC -> TS3K  
<STX>SF<CR><ETX>  
  
// RECEIVE : TE3K -> PC  
<STX>FL,(buf : 250 byte )<EOF><ETX>  
  
  
unsigned char buf[512]; // actual recieving data  size is 250 byte  
union {  
    float fData;  
    long  lData;  
    short sData[2];  
    unsigned char uc[4];  
} chgEndianData;  
  
  
// -----------------------------  
// (I) Motion Status : total 52 byte  
// -----------------------------  
// 1) Servo power status : 1byte  
robotData.status.servo  = buf[0];  
// 2) EMERGENCY stop switch status  : 1byte  
robotData.status.emgSW  = buf[1];  
// 3) Motion status   : 1byte  
//     0:  STOP (RESET)  
//     1:  RUN  
//     2:  STOP (RETRY)  
//     3:  STOP (CONT)  
robotData.status.status = buf[2];  
// 4) SU command request : 1byte  
// buf[3]  
  
// 5) Current alarm information : 2 byte \* 10  
pnt = 4;  
for (i = 0; i < 10; ++i, pnt += 2) {  
    chgEndianData.uc[1] = buf[pnt];  
    chgEndianData.uc[0] = buf[pnt+1];  
    robotData.status.alarm[i] = chgEndianData.sData[0];  
}  
  
// 6) Program execution line : 2 byte  
chgEndianData.uc[1] = buf[24];  
chgEndianData.uc[0] = buf[25];  
robotData.status.execLine = chgEndianData.sData[0];  
  
// 7) Program analysis line : 2 byte  
chgEndianData.uc[1] = buf[26];  
chgEndianData.uc[0] = buf[27];  
robotData.status.analLine = chgEndianData.sData[0];  
  
// 8) Program execution task : 2 byte  
chgEndianData.uc[1] = buf[28];  
chgEndianData.uc[0] = buf[29];  
robotData.status.execTask = chgEndianData.sData[0];  
  
// 9) Program analysis task : 2 byte  
chgEndianData.uc[1] = buf[30];  
chgEndianData.uc[0] = buf[31];  
robotData.status.analTask = chgEndianData.sData[0];  
  
// 10) Feed hold status : 2 byte  
chgEndianData.uc[1] = buf[32];  
chgEndianData.uc[0] = buf[33];  
robotData.status.feedHold = chgEndianData.sData[0];  
  
// 11) Guidance coordinate system status : 2 byte  
chgEndianData.uc[1] = buf[34];  
chgEndianData.uc[0] = buf[35];  
robotData.status.guidCoord = chgEndianData.sData[0];  
  
// 12) Guide rate status  : 2 byte  
chgEndianData.uc[1] = buf[36];  
chgEndianData.uc[0] = buf[37];  
robotData.status.guidRate = chgEndianData.sData[0];  
  
// 13) Guide mode status : 2 byte  
chgEndianData.uc[1] = buf[38];  
chgEndianData.uc[0] = buf[39];  
robotData.status.guidMode = chgEndianData.sData[0];  
  
// 14) Master mode status : 2 byte  
chgEndianData.uc[1] = buf[40];  
chgEndianData.uc[0] = buf[41];  
robotData.status.masterMode = chgEndianData.sData[0]; // TEACHING,INTERNAL,EXT(SIG),EXT(HOST) //  
// 15)  
// dummy : buf[42];  
// dummy : buf[43];  
  
// 16) power ON time : 2 byte  
chgEndianData.uc[3] = buf[44];  
chgEndianData.uc[2] = buf[45];  
chgEndianData.uc[1] = buf[46];  
chgEndianData.uc[0] = buf[47];  
robotData.status.powerOnTime = chgEndianData.lData;  
  
// 17) Program run time : 2 byte  
chgEndianData.uc[3] = buf[48];  
chgEndianData.uc[2] = buf[49];  
chgEndianData.uc[1] = buf[50];  
chgEndianData.uc[0] = buf[51];  
robotData.status.prgExecTime = chgEndianData.lData;  
  
// -----------------------------  
// (II) I/O information : total 64 byte  
// -----------------------------  
  
// 1) ~ 16) input  
memcpy(robotData.IO.input , &buf[52], 32);  
// 17) ~ 32) input  
memcpy(robotData.IO.output, &buf[84], 32);  
pnt = 116;  
  
// -----------------------------  
// (III) Current value data : total 134 byte  
// -----------------------------  
  
// 1)  Joint coordinate value : 4 byte \* 6  
for (i = 0; i < 6; ++i, pnt += sizeof(float)) {  
    chgEndianData.uc[3] = buf[pnt];  
    chgEndianData.uc[2] = buf[pnt+1];  
    chgEndianData.uc[1] = buf[pnt+2];  
    chgEndianData.uc[0] = buf[pnt+3];  
    robotData.position.joint[i] = chgEndianData.fData;  
}  
  
// 2)  World coordinate value : 4 byte \* 6  
for (i = 0; i < 6; ++i, pnt += sizeof(float)) {  // ワールド座標 //  
     chgEndianData.uc[3] = buf[pnt];  
     chgEndianData.uc[2] = buf[pnt+1];  
     chgEndianData.uc[1] = buf[pnt+2];  
     chgEndianData.uc[0] = buf[pnt+3];  
     robotData.position.world[i] = chgEndianData.fData;  
     }  
  
// 3) Work coordinate value : 4 byte \* 6  
for (i = 0; i < 6; ++i, pnt += sizeof(float)) {  // ワーク座標   //  
    chgEndianData.uc[3] = buf[pnt];  
     chgEndianData.uc[2] = buf[pnt+1];  
     chgEndianData.uc[1] = buf[pnt+2];  
     chgEndianData.uc[0] = buf[pnt+3];  
     robotData.position.work [i] = chgEndianData.fData;  
}  
// 4)  Work coordinate name : 20 byte  
memcpy(robotData.coord.workName, &buf[188], 20);  
// 5)  Tool coordinate name : 20 byte  
memcpy(robotData.coord.toolName, &buf[208], 20);  
// 6)  Base coordinate name : 20 byte  
memcpy(robotData.coord.baseName, &buf[228], 20);  
  
// 7)  Reserved : 2 byte  
// buf[229]  
// buf[230]