#define SENSOR\_PM2\_5\_TYP\_CLYLE

#include "Public.h"

void f\_PM2\_5Init(void)

{

f\_RingBufInit(&PlanTowerDataCur.PM2\_5RingBuffer);

}

 unsigned short f\_CheckSumPm2\_5(unsigned char \*a,unsigned char start,unsigned char lenbuf)

{

unsigned char i = 0;

unsigned short sumbuf = 0x0000;

for(i = start;i <= start+lenbuf;i++)

sumbuf += (unsigned short) a[i];

return sumbuf;

}

void f\_GetBagPm2\_5(PlanTowerData \* pmbuf)

{

if(f\_RingCanReadLen(&pmbuf->PM2\_5RingBuffer) == 0)

return;

if(!pmbuf->bPm2\_5ReOverFlag)

{

f\_ReadRing(&pmbuf->PM2\_5RingBuffer,&pmbuf->Pm2\_5CurData);

if((pmbuf->Pm2\_5LastData == PM2\_5\_HEAD1)&&(pmbuf->Pm2\_5CurData == PM2\_5\_HEAD2))    //头码

{

pmbuf->Pm2\_5ReDataBuf[0] = PM2\_5\_HEAD1;

pmbuf->Pm2\_5ReDataBuf[1] = PM2\_5\_HEAD2;

pmbuf->Pm2\_5RxCount = 2;

return;

}

pmbuf->Pm2\_5ReDataBuf[pmbuf->Pm2\_5RxCount] = pmbuf->Pm2\_5CurData;

pmbuf->Pm2\_5RxCount++;

if(pmbuf->Pm2\_5RxCount >= 100)

pmbuf->Pm2\_5RxCount = 0;

      pmbuf->Pm2\_5LastData = pmbuf->Pm2\_5CurData;

if(pmbuf->Pm2\_5RxCount == 4)

{

pmbuf->Pm2\_5RxLen = pmbuf->Pm2\_5ReDataBuf[2]\*256 + pmbuf->Pm2\_5ReDataBuf[3];

}

if(pmbuf->Pm2\_5RxCount == (pmbuf->Pm2\_5RxLen + 4))

{

if(pmbuf->Pm2\_5ReDataBuf[pmbuf->Pm2\_5RxLen + 2]\*256+pmbuf->Pm2\_5ReDataBuf[pmbuf->Pm2\_5RxLen+ 3]

!= f\_CheckSumPm2\_5(pmbuf->Pm2\_5ReDataBuf,0,pmbuf->Pm2\_5RxLen + 1))

return;

pmbuf->bPm2\_5ReOverFlag = TRUE;

memcpy((void \*)(&PM2\_5FrameCur),pmbuf->Pm2\_5ReDataBuf,pmbuf->Pm2\_5RxLen + 4);

pmbuf->Pm2\_5ErrTm = 0;

}

}

}

void f\_GetPm2\_5Data(PM2\_5Frame \*framebuf,PM2\_5Data \* databuf,PM2\_5\_STEP \*stepbuf)

{

if(PlanTowerDataCur.bPm2\_5ReOverFlag)

PlanTowerDataCur.bPm2\_5ReOverFlag = FALSE;

else

return;

databuf->PM1\_0Data = framebuf->PM1\_0[0]  \* 256 + framebuf->PM1\_0[1];

databuf->PM2\_5Data = framebuf->PM2\_5[0]  \* 256 + framebuf->PM2\_5[1];

databuf->PM10Data = framebuf->PM10[0] \* 256 + framebuf->PM10[1];

databuf->PM1\_0\_AIRData = framebuf->PM1\_0\_AIR[0]  \* 256 + framebuf->PM1\_0\_AIR[1];

databuf->PM2\_5\_AIRData = framebuf->PM2\_5\_AIR[0]  \* 256 + framebuf->PM2\_5\_AIR[1];

databuf->PM10\_AIRData = framebuf->PM10\_AIR[0]  \* 256 + framebuf->PM10\_AIR[1];

// databuf->PMHchoData = framebuf->PM\_HCHO[0]  \* 256 + framebuf->PM\_HCHO[1];

if(databuf->PM1\_0Data < PM2\_5\_STEP1\_VALUE)

{

stepbuf->PM1\_0Step = PM\_STEP1;

}

else if(databuf->PM1\_0Data < PM2\_5\_STEP2\_VALUE)

{

stepbuf->PM1\_0Step = PM\_STEP2;

}

else

{

stepbuf->PM1\_0Step = PM\_STEP3;

}

if(databuf->PM2\_5Data < PM2\_5\_STEP1\_VALUE)

{

stepbuf->PM2\_5Step = PM\_STEP1;

}

else if(databuf->PM2\_5Data < PM2\_5\_STEP2\_VALUE)

{

stepbuf->PM2\_5Step = PM\_STEP2;

}

else

{

stepbuf->PM2\_5Step = PM\_STEP3;

}

if(databuf->PM10Data < PM10\_STEP1\_VALUE)

{

stepbuf->PM10Step = PM\_STEP1;

}

else if(databuf->PM10Data < PM10\_STEP2\_VALUE)

{

stepbuf->PM10Step = PM\_STEP2;

}

else

{

stepbuf->PM10Step = PM\_STEP3;

}

}

/\*===================================================================

 \* 函数名称:  f\_DealSensorPm2\_5()

 \* 功能说明:  PM2.5数据处理

 \* 输入参数:

 \* 输出参数:

 \* 编辑时间:  2016-06-07

 \* 编辑作者:  王聪

 \* 备           注:

 \*==================================================================\*/

void f\_DealSensorPm2\_5(void)

{

f\_Pm2\_5Rst();

f\_GetBagPm2\_5(&PlanTowerDataCur);

f\_PmErrorDeal(&PlanTowerDataCur);

f\_PmErrorDeal(&PlanTowerDataCur);

f\_GetPm2\_5Data(&PM2\_5FrameCur,&PM2\_5DataCur,&PM2\_5\_StepCur);

}