

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>

//mk_mea_rom.c 總匯總力總
#define MEA1_GF0 5.0e-6
#define MEA1_GF1 1.0e-5
#define MEA1_GF2 2.0e-6
#define MEA1_GF3 5.0e-7
#define MEA1_GF4 2.0e-7

#define MEA2_GF0 1.0e-4
#define MEA2_GF1 1.5e-4
#define MEA2_GF2 1.5e-5
#define MEA2_GF3 7.0e-6
#define MEA2_GF4 2.5e-6

#define d_Lcl_MEA_MAX_V32 1.0273e5 //LOOKUP TABLE Velocity32 Normalization parameter (km/s) 30keV
#define d_Lcl_MEA_TBL_VF 0x27f//(d_Lcl_MEA_TBL_GF) //vf, median (1)
#define d_Lcl_MEA_TBL_GF 0xbf //gf&gf2, median (1)

char conv_VM_MEA(double*, double*, unsigned int, int, int, int);
char USHORTtoDBL(double*, unsigned short*, unsigned int);

int main(int argc, char *argv[])
{
    int i;
    long ltmp;

    unsigned short us_val[13];
    double d_val[13];
    double d_val_phys[13];

    if (argc != 14) {
        printf("Usage: ./***.exe 13 VM values (ex. 1111 2222 ... aaaa(hex))\n");
        exit(1);
    }

    for (i = 0; i < 13; ++i) us_val[i] = (unsigned short) strtol(argv[i], NULL, 16);

    USHORTtoDBL(d_val, us_val, 13);

    conv_VM_MEA(d_val_phys, d_val, 13, 0, 0, 1);

    for (i = 0; i < 13; ++i) printf("%.2e ", d_val_phys[i]);
    printf("\n");
}

```

```

char conv_VM_MEA( double f_VMout[],      //[OUTPUT]
                 double f_VMin[],      //[INPUT] f_VM[13]
                 unsigned int len,      //[INPUT] 13 values for MEA
                 int i_gf,              //[INPUT] G-factor# 0-4
                 int i_snsr,           //[INPUT] 0: MEA1, 1:MEA2
                 int i_all)            //[INPUT] 0: partial VM, 1:VM of whole energy
{
    double egf = 1.;                  //eff*gf 隱蝸綑綑綑綑綑
    double dEoverE = 1.;              //deltaE/E          [[ 隱蝸 ]]
    double EngtoVlc = 1.;             //sqrt(Energy)->velocity 隱蝸綑綑綑綑綑
    double V_MAX = 1.;               //sqrt(Energy) 隱蝸綑綑綑綑綑綑綑
    double dtheta = 1.;              //delta theta S/C 隱蝸綑綑綑 [[ 隱蝸 ]]
    double dazmth = 1.;             //delta azimth S/C 隱蝸綑綑綑 [[ 隱蝸 ]]
    double Time = 0.01;              //Sampling time      [[ 隱蝸 ]]

    EngtoVlc = sqrt(2*1.60218e-19/1.67262e-27)*100;    //(eV) -> (cm/s)

    if (i_snsr == 0) {
        dEoverE = 0.1;
        dtheta = 180./8/180*M_PI;
        dazmth = 180./8/180*M_PI;
        Time = 0.0078;
        V_MAX = d_LcL_MEA_MAX_V32*1e5;
        if (i_gf == 0) egf = MEA1_GF0;
        else if (i_gf == 1) egf = MEA1_GF1;
        else if (i_gf == 2) egf = MEA1_GF2;
        else if (i_gf == 3) egf = MEA1_GF3;
        else if (i_gf == 4) egf = MEA1_GF4;
    }
    if (i_snsr == 1) {
        dEoverE = 0.1;
        dtheta = 180./8/180*M_PI;
        dazmth = 180./8/180*M_PI;
        Time = 0.0078;
        V_MAX = d_LcL_MEA_MAX_V32*1e5;
        if (i_gf == 0) egf = MEA2_GF0;
        else if (i_gf == 1) egf = MEA2_GF1;
        else if (i_gf == 2) egf = MEA2_GF2;
        else if (i_gf == 3) egf = MEA2_GF3;
        else if (i_gf == 4) egf = MEA2_GF4;
    }

    if (i_all) {
        f_VMout[0] = f_VMin[ 0]/10*0xffff/egf/V_MAX*dEoverE*dtheta*dazmth/Time;
        f_VMout[1] = f_VMin[ 1]/10*0xffff/egf *dEoverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
        f_VMout[2] = f_VMin[ 2]/10*0xffff/egf *dEoverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
        f_VMout[3] = f_VMin[ 3]/10*0xffff/egf *dEoverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
        f_VMout[4] = f_VMin[ 4]/10*0xffff/egf*V_MAX*dEoverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
        f_VMout[5] = f_VMin[ 5]/10*0xffff/egf*V_MAX*dEoverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
        f_VMout[6] = f_VMin[ 6]/10*0xffff/egf*V_MAX*dEoverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
        f_VMout[7] = f_VMin[ 7]/10*0xffff/egf*V_MAX*dEoverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
        f_VMout[8] = f_VMin[ 8]/10*0xffff/egf*V_MAX*dEoverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
    }
}

```

```

f_VMout[9] = f_VMin[ 9]/10*0xffff/egf*V_MAX*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[10]= f_VMin[10]/100*0xffff/egf*V_MAX*V_MAX*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[11]= f_VMin[11]/100*0xffff/egf*V_MAX*V_MAX*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[12]= f_VMin[12]/100*0xffff/egf*V_MAX*V_MAX*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
}
else {
f_VMout[0] = f_VMin[ 0]*10*0xffff/egf*dEverE*dtheta*dazmth/Time;
f_VMout[1] = f_VMin[ 1]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[2] = f_VMin[ 2]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[3] = f_VMin[ 3]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[4] = f_VMin[ 4]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[5] = f_VMin[ 5]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[6] = f_VMin[ 6]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[7] = f_VMin[ 7]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[8] = f_VMin[ 8]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[9] = f_VMin[ 9]/10*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[10]= f_VMin[10]/100*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[11]= f_VMin[11]/100*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
f_VMout[12]= f_VMin[12]/100*0xffff/egf*dEverE*dtheta*dazmth/Time/d_LcL_MEA_TBL_VF*d_LcL_MEA_TBL_GF;
}
}

//For WINDOWS LITTLE ENDIAN      !!while the MDP com. shall be by the big-endian
// assuing:
//      double 64bit
//      USHORT 16bit
char USHORTtoDBL(double *pf_outbuf,      // [OUTPUT]
                unsigned short *pus_inbuf, // [INPUT] us_VM[10] in Little endian
                unsigned int len)        // [INPUT] 10 values for MIA
{
    int i;

    //螺迎菴
    union {
        double d;//1(sign) + 11(exponent) + 52(fraction)
        unsigned char uc[8];
    } un_dat;

    for(i = 0; i < len; i++){

        //1(sign) + 5(exponent) + 10(fraction) -->
        //      1(sign) + 11(exponent) + 52(fraction)
        //1 + 11 + 52
        if (*pus_inbuf == 0x7c00) { //+辟彝山?
            un_dat.uc[7] = 0x7f;
            un_dat.uc[6] = 0xf0;
            un_dat.uc[5] = 0;
            un_dat.uc[4] = 0;
            un_dat.uc[3] = 0;
            un_dat.uc[2] = 0;

```

```

    un_dat.uc[1] = 0;
    un_dat.uc[0] = 0;
}
else if (*pus_inbuf == 0xfc00) {          //-辟嶽嶽
    un_dat.uc[7] = 0xff;
    un_dat.uc[6] = 0xf0;
    un_dat.uc[5] = 0;
    un_dat.uc[4] = 0;
    un_dat.uc[3] = 0;
    un_dat.uc[2] = 0;
    un_dat.uc[1] = 0;
    un_dat.uc[0] = 0;
}
else if (*pus_inbuf == 0) {             //0
    un_dat.d = 0;
}
else {                                  //exp != 0 嶽嶽嶽
    if ((*pus_inbuf >> 14) & 0x01) un_dat.uc[7] = (*pus_inbuf >> 8 & 0xc0);          //exp > 0x0f
    else un_dat.uc[7] = (*pus_inbuf >> 8 & 0xc0) | 0x3f; //exp <= 0x0f
    un_dat.uc[6] = (((*pus_inbuf >> 8 & 0x3f) << 2) | (((*pus_inbuf & 0xff) >> 6) & 0x03) );
    un_dat.uc[5] = (((*pus_inbuf & 0xff) & 0x3f) << 2);
    un_dat.uc[4] = 0;
    un_dat.uc[3] = 0;
    un_dat.uc[2] = 0;
    un_dat.uc[1] = 0;
    un_dat.uc[0] = 0;
}
*pf_outbuf++ = un_dat.d;
pus_inbuf++;

}
return 1;
}

```