MMO MPPE-MEA moment calculation

1. EEPROM Table preparation

2. Onboard calculation of VM

3. Conversion to physical amounts

1. EEPROM Table preparation

1.1. Protocol

[Prepare input files]

1. "./mea/MEA1\_FM\_MCP\_Rel\_Eff.txt" (Anode Normalized Efficiency)
2. "../mea/Mea1\_FM\_Properties\_1000eV\_03Jul\_2012\_A.txt"; (#G Utop/Uan #Anode G,cm2\_sr\_eV/eV Azimuth,deg Elevation,deg Elevation\_FWHM,deg K E/E0\_FWHM)

[Comment in/out “#define \_CONDUCT\_MEA1\_” for MEA1/2]

[Compile] gcc –o mk\_mea\_rom mk\_mea\_rom.c

[EXECUTE] ./mk\_mea\_rom.exe (without parameters)

1.2. Description of mk\_mea\_rom.c

・G-factor, MCP efficiency

* from G[j][i] (cm2\_sr\_eV/eV) and eff[i] (Normalized Efficiency), j = #G, i = #Anode.
* us\_gf[j][i] = 1/G[j][i]/eff[i] \* MEA1/2\_GF(0-4)\*d\_Lcl\_MEA\_TBL\_GF
* uc\_gf[j][i] = us\_gf[j][i] - d\_Lcl\_MEA\_TBL\_GF0 (for packing into 1B array)

 uc\_gf[j][10-13]= 0 of MEA1 due to the blind channels

**OUTPUT**>> uc\_gf[j][i] (1B x 5g-facotr x 16ch = 80B)

・THETA, PHI

* from AZ[j][i] (Azimuth,deg) and EL[j][i] (Elevation,deg.)
* theta[j][0-7] = AZ -90

theta[j][8-15] = -AZ -90

* phi[j][0-7] = EL

phi[j][8-15] = -EL

* coordinate transform
* c\_th[j][i] = (theta[j][i] - theta0[i]) \*M\_PI/180./d\_Lcl\_MEA\_MAX\_RAD\*d\_Lcl\_MEA\_TBL\_RAD

 theta0: nominal values

 (c\_th indicates difference from the nominal values to save the EEPROM space.)

* c\_pe[j][i] = phi[j][i] \*M\_PI/180./d\_Lcl\_MEA\_MAX\_RAD\*d\_Lcl\_MEA\_TBL\_RAD

**OUTPUT**>> c\_th[j][i] (1B x 5g-facotr x 16ch = 80B)

**OUTPUT**>> uc\_pe[j][i] (1B x 5g-facotr x 16ch = 80B)

・VF: Velocity CH-factor

* from K[j][i] (analyzer constant?)
* V[j][i] = sqrt(K[j][i]), V0 = sqrt(MEA\_ANLZR\_CNST0)
* uc\_vc[j][i] = (V[j][i] - V0)/V0 \*d\_Lcl\_MEA\_TBL\_VF + 0x7f;

 V0： Standard

 + 0x7f;　to be unsigned char

**OUTPUT**>> uc\_vc[j][i] (1B x 5g-facotr x 16ch = 80B)

・VELOCITY

* from Energy table list
* void input\_vtable(USHORT us\_v32[]): Energy table128 -> velocity table32
* V = v[T1-3][32](km/s) /d\_Lcl\_MEA\_MAX\_V32\*0xffff

**OUTPUT**>>T1-3 Velocity (32en x3 = 192B)

/mk\_mea\_rom.exe should run on WINDOWS(little endian), thus in the function of OUTPUT, byte swap is done for big-endian MDP.

2. SW calculation by MDP SW

2.1. Read parameters from EEPROM

app02\_MPPE-MEA.c

GF: pP->uc\_gf <<1/G/eff \* MEA1/2\_GF(0-4)\*d\_Lcl\_MEA\_TBL\_GF - d\_Lcl\_MEA\_TBL\_GF0

cos/sin(polar):

 c\_th[j][i] << (theta[j][i] - theta0[i])\*M\_PI/180./d\_Lcl\_MEA\_MAX\_RAD\*d\_Lcl\_MEA\_TBL\_RAD

 polar = pP-> c\_th[j][i] \* d\_Lcl\_MEA\_MAX\_RAD/d\_Lcl\_MEA\_TBL\_RAD + polar0 (rad.)

 theta0, polar0: nominal values

 pP->c\_csp,snp(i) = ( cos/sin(polar) - cos/sin(polar0) )\*d\_Lcl\_MEA\_TBL\_RD1

cos/sin(azmth):

 c\_az[j][i] << phi[j][i] \*M\_PI/180./d\_Lcl\_MEA\_MAX\_RAD\*d\_Lcl\_MEA\_TBL\_RAD

 azmth = pP->c\_az \*d\_Lcl\_MEA\_MAX\_RAD/d\_Lcl\_MEA\_TBL\_RAD + azmth0 (rad.)

 azmth0: nominal values

 pP->s\_csa,sna(i,j,k) = ( cos/sin(azmth) - cos/sin(azmth0) )\*d\_Lcl\_MEA\_TBL\_RD1

VLC: pP->us\_vlc << V ( v\*0xffff/d\_Lcl\_MEA\_MAX\_V32)

 pP->uc\_vf << uc\_vc[j][i] ( +0x200 makes it ( 0x27f +- 0x7f ) later. )

2.2. VM calculation

app03\_MPPE-MEA.c

* us\_gf2 = d\_Lcl\_MEA\_TBL\_GF0 + pP->uc\_gf2
* us\_vf = d\_LCl\_MEA\_TBL\_VF0(0x200) + pP->uc\_vf
* s\_cosp/sinp = pP->c\_csp/snp + pP->s\_csp0/snp0
* s\_cosa/sina = pP->c\_csp/snp + pP->s\_csp0/snp0
* S/C potential:

 A/B = 100\*(cmd\_val\_A/B [0-0xfff] - 0x7ff)/0x7ff (-100 -- +100)

 S/C\_pot = A\* Gs\_TRG2\_EFD\_Pot[i]\*d\_Lcl\_MEA\_MAX\_POT/0x7fff + B (vol.)

 S/C\_v = sqrt(2q/m)\*sqrt(S/C\_pot) (km/s)

 v = pP->us\_vlc[k]\*0xffff/d\_Lcl\_MEA\_MAX\_V32

 correction factor: v' = sqrt(1 - S/C\_v^2/v^2) \* d\_Lcl\_MEA\_TBL\_POT

 serach k#: > S/C\_v\*sqrt(2) > v[k] k >> i\_en\_pot (to be used as MV calculation limit)

**All**

n = count/VLC\* v'\*GF\*cos 0.1/( TBL\_POT\*TBL\_RD1\*TBL\_GF/TBL\_VF)

Vx = count \* v'^2\*GF\*cos\*cos\*cos 10/( 0xffff\*TBL\_POT^2\*TBL\_RD1^3\*TBL\_GF^2/TBL\_VF)

Vy = count \* v'^2\*GF\*cos\*cos\*sin 10/( 0xffff\*TBL\_POT^2\*TBL\_RD1^3\*TBL\_GF^2/TBL\_VF)

Vz = count \* v'^2\*GF\*cos\*sin 10/( 0xffff\*TBL\_POT^2\*TBL\_RD1^2\*TBL\_GF^2/TBL\_VF)

Pxx=count\*VLC\*v'^3\*GF\*cos\*cos\*cos\*cos\*cos 10/( 0xffff^2\*TBL\_POT^3\*TBL\_RD1^5\*TBL\_GF^2)

Pyy = count\*VLC\*v'^3\*GF\*cos\*cos\*cos\*sin\*sin 10/( 0xffff^2\*TBL\_POT^3\*TBL\_RD1^5\*TBL\_GF^2)

Pzz = count\*VLC\*v'^3\*GF\*cos\*sin\*sin 10/( 0xffff^2\*TBL\_POT^3\*TBL\_RD1^3\*TBL\_GF^2)

Pxy = count\*VLC\*v'^3\*GF\*cos\*cos\*cos\*cos\*sin 10/( 0xffff^2\*TBL\_POT^3\*TBL\_RD1^5\*TBL\_GF^2)

Pyz = count\*VLC\*v'^3\*GF\*cos\*cos\*sin\*sin 10/( 0xffff^2\*TBL\_POT^3\*TBL\_RD1^4\*TBL\_GF^2)

Pxz = count\*VLC\*v'^3\*GF\*cos\*cos\*sin\*cos 10/( 0xffff^2\*TBL\_POT^3\*TBL\_RD1^4\*TBL\_GF^2)

qx=count\*VLC^2\*v'^4\*GF\*cos\*cos\*cos 100/( 0xffff^3\*TBL\_POT^4\*TBL\_RD1^3\*TBL\_GF^2 \*TBL\_VF)

qy=count\*VLC^2\*v'^4\*GF\*cos\*cos\*sin 100/( 0xffff^3\*TBL\_POT^4\*TBL\_RD1^3\*TBL\_GF^2 \*TBL\_VF)

qz=count\*VLC^2\*v'^4\*GF\*cos\*sin 100/( 0xffff^3\*TBL\_POT^4\*TBL\_RD1^2\*TBL\_GF^2 \*TBL\_VF)

 \*)v' of factor is added according to MEA CDR package p 31

**Partial (single energy)**

n = count\*GF\*cos 0.1/( 0xffff\*TBL\_RD1\*TBL\_GF/TBL\_VF)

Vx = count\*GF\*cos\*cos\*cos 10/( 0xffff\*TBL\_RD1^3\*TBL\_GF^2/TBL\_VF)

Vy = count\*GF\*cos\*cos\*sin 10/( 0xffff\*TBL\_RD1^3\*TBL\_GF^2/TBL\_VF)

Vz = count\*GF\*cos\*sin 10/( 0xffff\*TBL\_RD1^2\*TBL\_GF^2/TBL\_VF)

Pxx=count\*GF\*cos\*cos\*cos\*cos\*cos 10/( 0xffff\*TBL\_RD1^5\*TBL\_GF^2)

Pyy = count\*GF\*cos\*cos\*cos\*sin\*sin 10/( 0xffff\*TBL\_RD1^5\*TBL\_GF^2)

Pzz = count\*GF\*cos\*sin\*sin 10/( 0xffff\*TBL\_RD1^3\*TBL\_GF^2)

Pxy = count\*GF\*cos\*cos\*cos\*cos\*sin 10/( 0xffff\*TBL\_RD1^5\*TBL\_GF^2)

Pyz = count\*GF\*cos\*cos\*sin\*sin 10/( 0xffff\*TBL\_RD1^4\*TBL\_GF^2)

Pxz = count\*GF\*cos\*cos\*sin\*cos 10/( 0xffff\*TBL\_RD1^4\*TBL\_GF^2)

qx=count\*GF\*cos\*cos\*cos 100/( 0xffff\* TBL\_RD1^3\*TBL\_GF^2 \*TBL\_VF)

qy=count\*GF\*cos\*cos\*sin 100/( 0xffff\* TBL\_RD1^3\*TBL\_GF^2 \*TBL\_VF)

qz=count\*GF\*cos\*sin 100/( 0xffff\* TBL\_RD1^2\*TBL\_GF^2 \*TBL\_VF)

Dimension after normalization

**All**

n = count/VLC\*gf\*0.1

Vx = count \*gf/(0xffff)\*10\*TBL\_VF/TBL\_ GF

Vy = count \*gf/(0xffff)\*10\*TBL\_VF/TBL\_ GF

Vz = count \*gf/(0xffff)\*10\*TBL\_VF/TBL\_ GF

Pxx = count\*VLC\*gf/(0xffff)^2\*10\*TBL\_VF/TBL\_GF

Pyy = count\*VLC\*gf/(0xffff)^2\*10\*TBL\_VF/TBL\_GF

Pzz = count\*VLC\*gf/(0xffff)^2\*10\*TBL\_VF/TBL\_GF

Pxy = count\*VLC\*gf/(0xffff)^2\*10\*TBL\_VF/TBL\_GF

Pyz = count\*VLC\*gf/(0xffff)^2\*10\*TBL\_VF/TBL\_GF

Pxz = count\*VLC\*gf/(0xffff)^2\*10\*TBL\_VF/TBL\_GF

qx = count\*VLC^2\*gf/(0xffff)^3\*100\*TBL\_VF/TBL\_GF

qy = count\*VLC^2\*gf/(0xffff)^3\*100\*TBL\_VF/TBL\_GF

qz = count\*VLC^2\*gf(0xffff)^3\*100\*TBL\_VF/TBL\_GF

**Partial (single energy)**

n = count/0xffff\*gf\*0.1

Vx = count/0xffff\*gf\*10\*TBL\_VF/TBL\_ GF

Vy = count/0xffff\*gf\*10\*TBL\_VF/TBL\_ GF

Vz = count/0xffff\*gf\*10\*TBL\_VF/TBL\_ GF

Pxx = count/0xffff\*gf\*10\*TBL\_VF/TBL\_GF

Pyy = count/0xffff\*gf\*10\*TBL\_VF/TBL\_GF

Pzz = count/0xffff\*gf\*10\*TBL\_VF/TBL\_GF

Pxy = count/0xffff\*gf\*10\*TBL\_VF/TBL\_GF

Pyz = count/0xffff\*gf\*10\*TBL\_VF/TBL\_GF

Pxz = count/0xffff\*gf\*10\*TBL\_VF/TBL\_GF

qx = count/0xffff\*gf\*100\*TBL\_VF/TBL\_GF

qy = count/0xffff\*gf\*100\*TBL\_VF/TBL\_GF

qz = count/0xffff\*gf\*100\*TBL\_VF/TBL\_GF

3. Conversion

conv\_MIA\_VM.c

3.1. 16bit unsigned short to 64bit float conv.

Done by char USHORTtoDBL(double\*, unsigned short\*, unsigned int) on the WINDOWS(LITTLE ENDIAN) in which parameters of USHORT are prepared in Little endian.

3.2. Conv. to physical parameters

Ex.) Done by char conv\_VM(double\*, double\*, unsigned int, int)

Prepare the following:

dE\_E: deltaEnergy/Energy

dp: delta\_polar (radian)

da: delta\_azimuth (radian)

T: Sampling time (s)

**All**

n \*= /0.1/MEA#\_GF# /(MEA\_MAX\_V32\*1e5/0xffff) \*dE\_E\*dp\*da/T

 \*= 10\*0xffff/MIN\_GF\_#/(MEA\_MAX\_V32\*1e5) \*dE\_E\*dp\*da/T

(ions/cm^3)

nV\*=/10\* 0xffff/MIN\_GF\_# \*dE\_E\*dp\*da/T /TBL\_VF\*TBL\_GF

 (ions/cm^2 sec)

P \*=/10\*0xffff^2/MIN\_GF\_# \*MEA\_MAX\_V32\*1e5/0xffff \*dE\_E\*dp\*da/T /TBL\_VF\*TBL\_GF

 \*= 0.1\*0xffff /MIN\_GF\_# \*MEA\_MAX\_V32\*1e5 \*dE\_E\*dp\*da/T /TBL\_VF\*TBL\_GF

 (ions/cm sec^2)

q \*= /100\*0xffff^3/MIN\_GF\_# \*(MEA\_MAX\_V32\*1e5/0xffff)^2\*dE\_E\*dp\*da/T /TBL\_VF\*TBL\_GF

 \*=0.01\*0xffff /MIN\_GF\_# \*(MEA\_MAX\_V32\*1e5)^2 \*dE\_E\*dp\*da/T /TBL\_VF\*TBL\_GF

 (ions/sec^3)

**Partial (single energy)**

n \*= /0.1\*0xffff/MEA#\_GF# \*dE\_E\*dp\*da/T

nV\*= /10\* 0xffff/MIN\_GF\_# \*dE\_E\*dp\*da/T /TBL\_VF\*TBL\_GF

P \*= /10\*0xffff/MIN\_GF\_# \*dE\_E\*dp\*da/T /TBL\_VF\*TBL\_GF

q \*= /100\*0xffff/MIN\_GF\_# \*dE\_E\*dp\*da/T /TBL\_VF\*TBL\_GF