

---

```

/*****
*/
*/
/*  MprgStruct.h : Mercury Micro Program Registers definition */
*/
*/
/*****
*/
*/
*/
*/
*/
*/
*/
/***** Copyright (C) Yaskawa Electric Corporation *****/
/*
/*  2014.02.19 : Made by Excel Macro */
/*
*/
/*****
#include "Basedef.h"
#include "SvIpAccess.h"

/*****
*/
/*  defines for Micro program */
/*
/*****
/***** Major Constant Values */
/*****
#define ZEROR    0x00000000    /* "0" */
#define ONER     0x00000001    /* "1" */
#define NONER    0xffffffff    /* "-1" */

```

---

```

/*****
/*  ServoIP Memory Maps */
/*****
#define SVIP_TOP_ADR 0x600 /* H/Wレジスタ先頭アドレス */
#define SVIP_AXIS_SIZE 0x100 /* 1軸当たりの使用領域 */

/*****
/*  degree value */
/*****
#define PI2 16384 /* 90deg */
#define PI23 21845 /* 120deg */

/*****
/*  AMPTYPE：回転型 / リニア型選択 */
/*****
#define ROTARY 0x0000 /* 回転型 */
#define LINEAR 0x0001 /* リニア型 */

/*****
/*  PAOSEQCMD：PaoSeqMcnd Interface Definition */
/*****
#define PAONOCMD 0x0000 /* No Command (Host CPU:KPX_PAOSEQ_NOCMD) */
#define PAOPLSOUT 0x0020 /* IncPulse Output Request (Host CPU:KPX_IPLSOUT_REQUEST) */

/*****
/*  POSETCMD：パルス出力回路初期化要求 (絶対値授受シーケンスのパルス出力回路初期化用) */
/*****
#define POSETNOCMD 0x0000 /* 要求なし */
#define POSETCMD00 0x0001 /* パルス変換位置に 0 0 0 0 h を強制セット要求 */
#define POSETCMDFF 0x0002 /* パルス変換位置に F F F F h を強制セット要求 */

/*****

```

---

---

```

/* CPORT read data */
/*****
//;-- 0x0001 /* bit.0 : --Reserve-- */
//;-- 0x0002 /* bit.1 : --Reserve-- */
//;-- 0x0004 /* bit.2 : --Reserve-- */
#define RLOCK 0x0008 /* bit.3 : Transer register lock status */
#define BB 0x0010 /* bit.4 : Base block bit */
//add tanaka21,2013.10.28
//;-- 0x0001 /* bit.0 : --Reserve-- */
//;-- 0x0002 /* bit.1 : --Reserve-- */
//;-- 0x0004 /* bit.2 : --Reserve-- */
#define RLOCK1 0x0008 /* bit.3 : Transer register lock status */
#define BB1 0x0010 /* bit.4 : Base block bit */
//;-- 0x0020 /* bit.5 : --Reserve-- */
//;-- 0x0040 /* bit.6 : --Reserve-- */
//;-- 0x0080 /* bit.7 : --Reserve-- */
//;-- 0x0100 /* bit.8 : --Reserve-- */
//;-- 0x0200 /* bit.9 : --Reserve-- */
//;-- 0x0400 /* bit.10: --Reserve-- */
#define RLOCK2 0x0800 /* bit.11: Transer register lock status */
#define BB2 0x1000 /* bit.12: Base block bit */
//;-- 0x2000 /* bit.13: --Reserve-- */
//;-- 0x4000 /* bit.14: --Reserve-- */
//;-- 0x8000 /* bit.15: --Reserve-- */

/*****
/* status set */
/*****
//;-- 0x0001 /* bit.0 : --Reserve-- */
//;-- 0x0002 /* bit.1 : --Reserve-- */
//;-- 0x0004 /* bit.2 : --Reserve-- */
#define DLIM 0x0008 /* bit.3 : D-axis integration limit status */
#define QLIM 0x0008 /* bit.3 : Q-axis integration limit status */
//;-- 0x0010 /* bit.4 : --Reserve-- */
#define TLIM 0x0020 /* bit.5 : Torque Limit status */
#define IHOLD 0x0040 /* bit.6 : Anti-windup test */

```

---

---

```

//;--      0x0080      /* bit.7 : --Reserve--      */
//add tanaka21,2013.10.28
//;--      0x0001      /* bit.0 : --Reserve--      */
//;--      0x0002      /* bit.1 : --Reserve--      */
//;--      0x0004      /* bit.2 : --Reserve--      */
#define DLIM1 0x0008      /* bit.3 : D-axis integration limit status      */
#define QLIM1 0x0008      /* bit.3 : Q-axis integration limit status      */
//;--      0x0010      /* bit.4 : --Reserve--      */
#define TLIM1 0x0020      /* bit.5 : Torque Limit status      */
#define IHOLD1 0x0040      /* bit.6 : Anti-windup test      */
//;--      0x0080      /* bit.7 : --Reserve--      */
//;--      0x0100      /* bit.8 : --Reserve--      */
//;--      0x0200      /* bit.9 : --Reserve--      */
//;--      0x0400      /* bit.10: --Reserve--      */
#define DLIM2 0x0800      /* bit.11: D-axis integration limit status      */
#define QLIM2 0x0800      /* bit.11: Q-axis integration limit status      */
//;--      0x1000      /* bit.12: --Reserve--      */
#define TLIM2 0x2000      /* bit.13: Torque Limit status      */
#define IHOLD2 0x4000      /* bit.14: Anti-windup test      */
//;--      0x8000      /* bit.15: --Reserve--      */

```

```

/*****
/*      status clear      */
*****/
//;--      0xffffe      /* bit.0 : --Reserve--      */
//;--      0xffffd      /* bit.1 : --Reserve--      */
//;--      0xffffb      /* bit.2 : --Reserve--      */
#define DLIMI 0xffff7      /* bit.3 : D-axis integration limit status      */
#define QLIMI 0xffff7      /* bit.3 : Q-axis integration limit status      */
//;--      0xfffff      /* bit.4 : --Reserve--      */
#define TLIMI 0xffdf      /* bit.5 : Torque Limit status      */
#define IHOLDI 0xffbf      /* bit.6 : Anti-windup test      */
//;--      0xff7f      /* bit.7 : --Reserve--      */
//add tanaka21,2013.10.28
//;--      0xffffe      /* bit.0 : --Reserve--      */
//;--      0xffffd      /* bit.1 : --Reserve--      */

```

```

//;--      0xffffb      /* bit.2 : --Reserve--      */
#define DLIMI1      0xffff7      /* bit.3 : D-axis integration limit status      */
#define QLIMI1      0xffff7      /* bit.3 : Q-axis integration limit status      */
//;--      0xffef      /* bit.4 : --Reserve--      */
#define TLIMI1      0xffdf      /* bit.5 : Torque Limit status      */
#define IHOLDI1      0xffbf      /* bit.6 : Anti-windup test      */
//;--      0xff7f      /* bit.7 : --Reserve--      */
//;--      0xfeff      /* bit.8 : --Reserve--      */
//;--      0xfdff      /* bit.9 : --Reserve--      */
//;--      0xfbff      /* bit.10: --Reserve--      */
#define DLIMI2      0xf7ff      /* bit.11: D-axis integration limit status      */
#define QLIMI2      0xf7ff      /* bit.11: Q-axis integration limit status      */
//;--      0xffff      /* bit.12: --Reserve--      */
#define TLIMI2      0xdfff      /* bit.13: Torque Limit status      */
#define IHOLDI2      0xbfff      /* bit.14: Anti-windup test      */
//;--      0x7fff      /* bit.15: --Reserve--      */

/*****
/*      CSW (Control Select Switch )      */
/*****
#define ICLR      0x0001      /* bit.0 積 分 リ ミ ッ ト 時 処 理 : ホ ー ル ド ( 0 ) / ク リ ア ( 1 )      */
#define ISEL      0x0002      /* bit.1 Ld/Lq補 償 : 電 流 F B ( 0 ) / 電 流 指 令 ( 1 )      */
//;--      0x0004      /* bit.2 --Reserve--      */
#define OBSEL      0x0008      /* bit.3 Current Observer Select bit      */
#define F1DSABL      0x0010      /* bit.4 Notch filter1 Disable      */
#define F2DSABL      0x0020      /* bit.5 Notch filter2 Disable      */
#define LPFDSABL      0x0040      /* bit.6 Low Pass filter1 Disable      */
#define LPFCDSABL      0x0080      /* bit.7 Low Passfilter2 Disable      */
#define OVMSSEL1      0x0100      /* bit.8 if bit8,9 = 01 then OVMOD1      */
#define OVMSSEL2      0x0200      /* bit.9 if bit8,9 = 10 or 11 then OVMOD2      */
#define OVMMOD      0x0400      /* bit.10 modulation modification enable      */
#define V_FB      0x0800      /* bit.11 電 圧 F B 方 式 弱 め 界 磁 選 択      */
#define DIDTSEL      0x1000      /* bit.12 L di/dt cmp select (1:di/dt,0:def)      */
#define V_FB2      0x2000      /* bit.13 電 圧 F B 方 式 弱 め 界 磁 選 択 2      */
#define INT_ST      0x4000      /* bit.14 飽 和 時 積 分 停 止      */
#define F3DSABL      0x8000      /* bit.15 Notch filter3 Disable      */

```

---

```

/*****
/*   CSW2 (Control Select Switch2 )                               */
/*****
#define F4DSABL  0x0001    /* bit.0  Notch filter4 Disable          */
#define F5DSABL  0x0002    /* bit.1  Notch filter5 Disable          */
#define F6DSABL  0x0004    /* bit.2  Notch filter6 Disable          */
//;--      0x0008    /* bit.3  --Reserve--                */
//;--      0x0010    /* bit.4  --Reserve--                */
//;--      0x0020    /* bit.5  --Reserve--                */
//;--      0x0040    /* bit.6  --Reserve--                */
//;--      0x0080    /* bit.7  --Reserve--                */
//;--      0x0100    /* bit.8  --Reserve--                */
//;--      0x0200    /* bit.9  --Reserve--                */
//;--      0x0400    /* bit.10 --Reserve--               */
//;--      0x0800    /* bit.11 --Reserve--               */
//;--      0x1000    /* bit.12 --Reserve--               */
//;--      0x2000    /* bit.13 --Reserve--               */
//;--      0x4000    /* bit.14 --Reserve--               */
//;--      0x8000    /* bit.15 --Reserve--               */

/*****
/*   ビ ッ ト パ ラ メ ー タ   (  初 期 化 時 セ ッ ト  C  P U → A S I C  )   */
/*****
#define UPGDIVOUT 0x0001    /* bit.0   マ イ ク ロ 分 周 機 能 使 用 選 択          */
//;--      0x0002    /* bit.1  --Reserve--                */
//;--      0x0004    /* bit.2  --Reserve--                */
//;--      0x0008    /* bit.3  --Reserve--                */
//;--      0x0010    /* bit.4  --Reserve--                */
//;--      0x0020    /* bit.5  --Reserve--                */
//;--      0x0040    /* bit.6  --Reserve--                */
//;--      0x0080    /* bit.7  --Reserve--                */
//;--      0x0100    /* bit.8  --Reserve--                */
//;--      0x0200    /* bit.9  --Reserve--                */
//;--      0x0400    /* bit.10 --Reserve--               */

```

---

---

```

//;--      0x0800      /* bit.11 --Reserve--          */
//;--      0x1000      /* bit.12 --Reserve--          */
//;--      0x2000      /* bit.13 --Reserve--          */
//;--      0x4000      /* bit.14 --Reserve--          */
//;--      0x8000      /* bit.15 --Reserve--          */

```

```

/*****
/*   ビ ッ ト デ ー タ   (   初 期 化 時 セ ッ ト   C P U → A S I C、ス キ ャ ン 処 理 A S I C → C P U   )   */
/*****
#define SERXERR      0x0001      /* bit.0  Senc Receive Error          */
#define ACCCHKENA    0x0002      /* bit.1  Encoder Acc. Check Enable Flag          */
//;--      0x0004      /* bit.2  --Reserve--          */
//;--      0x0008      /* bit.3  --Reserve--          */
//;--      0x0010      /* bit.4  --Reserve--          */
//;--      0x0020      /* bit.5  --Reserve--          */
//;--      0x0040      /* bit.6  --Reserve--          */
//;--      0x0080      /* bit.7  --Reserve--          */
//;--      0x0100      /* bit.8  --Reserve--          */
//;--      0x0200      /* bit.9  --Reserve--          */
//;--      0x0400      /* bit.10 --Reserve--          */
//;--      0x0800      /* bit.11 --Reserve--          */
//;--      0x1000      /* bit.12 --Reserve--          */
//;--      0x2000      /* bit.13 --Reserve--          */
//;--      0x4000      /* bit.14 --Reserve--          */
//;--      0x8000      /* bit.15 --Reserve--          */

```

```

/*****
/*                                     */
/*   Work Register : WREGBASE address = 08007000h          */
/*                                     */
/*****
/*-----*/
/*   Axis information register          */
/*-----*/

```

---

```

typedef struct _AXIS_INFO {
    LONG AxisNum;          /* 0xC0007000 : 制 御 軸数          */
} AXIS_INFO;

/*-----*/
/*    32 bit registers          */
/*-----*/
typedef struct _ACRV {
    DWREG IdIntgl;         /* 0xC0007000 : ACRd Integral          */
    DWREG IqIntgl;         /* 0xC0007004 : ACRq Integral          */
    DWREG VdFil;           /* 0xC0007008 : ACRd Filter Output      */
    DWREG VqFil;           /* 0xC000700C : ACRq Filter Output      */
} ACRV;

/*-----*/
/*    16 bit registers          */
/*-----*/
typedef struct _STSFLG {
    SHORT BbSetW;          /* 0xC0007010 : soft BB set(BBSET)      */
    SHORT FltStsW;         /* 0xC0007012 : fault status(FLTIN)     */
    SHORT CtrlStsRW;       /* 0xC0007014 : controll flag/status(CTSTR, CTSTW) */
    SHORT CNTSTS;          /* 0xC0007016 : counter status(FBCSTS)   */
    SHORT FccStsMon;       /* 0xC0007018 : full closed counter status(FCCST) */
    SHORT IntglFlg;        /* 0xC000701A : 積 分 飽 和 フラグ      */
} STSFLG;

/*-----*/
/*    voltage FF          */
/*-----*/
typedef struct _VCMPIV {
    SHORT LdC;             /* 0xC000701C : Ld * omega              */
    SHORT LqC;             /* 0xC000701E : Lq * omega              */
    SHORT MagC;            /* 0xC0007020 : Phi * omega             */
    SHORT VdOut;           /* 0xC0007022 : Vd output voltage        */
    SHORT VqOut;           /* 0xC0007024 : Vq output voltage        */
    SHORT VdComp;          /* 0xC0007026 : Vd compensation data     */
}

```

---



---

```

SHORT VqComp;      /* 0xC0007028 : Vq compensation data          */
SHORT VuOut;       /* 0xC000702A : vu output data (to PWM)                    */
SHORT VvOut;       /* 0xC000702C : vv output data (to PWM)                    */
SHORT VwOut;       /* 0xC000702E : vw output data (to PWM)                    */
/*-----*/
SHORT VdRef;       /* 0xC0007030 : vdref                                     */
SHORT VqRef;       /* 0xC0007032 : vqref                                     */
SHORT Vmax2;       /* 0xC0007034 : 電圧指令ベクトル最大値(8 192*1.27)        */
SHORT V12;         /* 0xC0007036 : 電圧指令ベクトル( $\sqrt{VdOut^2+VqOut^2}$ ) */
} VCMPIV;

/*-----*/
/* trigonometric function value */
/*-----*/
typedef struct _SINTBL {
    SHORT SinT;      /* 0xC0007038 : sin theta */
    SHORT CosT;      /* 0xC000703A : cos theta */
    SHORT SinT2;     /* 0xC000703C : sin(theta + 2pi/3) */
    SHORT CosT2;     /* 0xC000703E : cos(theta + 2pi/3) */
    SHORT SinT3;     /* 0xC0007040 : sin(theta - 2pi/3) */
    SHORT CosT3;     /* 0xC0007042 : cos(theta - 2pi/3) */
} SINTBL;

/*-----*/
/* A/D Stop Error detect */
/*-----*/
typedef struct _ADSTOP {
    SHORT ADSt;      /* 0xC0007044 : */
    SHORT ADERRCNT;  /* 0xC0007046 : ASIC AD Error Counter */
    SHORT IRIUP;     /* 0xC0007048 : Previous IRIU */
    SHORT IRIUCNT;   /* 0xC000704A : same IRIU counter */
    SHORT IRIVP;     /* 0xC000704C : Previous IRIV */
    SHORT IRIVCNT;   /* 0xC000704E : same IRIV counter */
} ADSTOP;

/*-----*/
/* CPU I/F Data */
/*-----*/

```

---

---

```

/*-----*/
typedef struct _ADINV {
    SHORT IuOffsetIn;    /* 0xC0007050 : A/D transfer offset for iu    */
    SHORT IvOffsetIn;    /* 0xC0007052 : A/D transfer offset for iv    */
    SHORT KcuIn;         /* 0xC0007054 : A/D transfer gain for iu      */
    SHORT KcvIn;         /* 0xC0007056 : A/D transfer gain for iv      */
    SHORT IdIn;          /* 0xC0007058 : Id reference                  */
    SHORT IqIn;          /* 0xC000705A : Iq reference                  */
    SHORT PhaseHIn;      /* 0xC000705C : PHA input                    */
    SHORT VelIn;         /* 0xC000705E : VEL input                    */
/*-----*/
/*    voltage compensation                */
/*-----*/
    SHORT VdRefIn;       /* 0xC0007060 : vdref input                  */
    SHORT VqRefIn;       /* 0xC0007062 : vqref input                  */
/*-----*/
/*    Torque Limit                        */
/*-----*/
    SHORT TLimPIn;       /* 0xC0007064 : +tlim                        */
    SHORT TLimMin;       /* 0xC0007066 : -tlim                        */
} ADINV;

/*-----*/
/*    static variable                    */
/*-----*/
typedef struct _INTADP {
    SHORT Kcu;           /* 0xC0007068 : A/D transfer gain for iu      */
    SHORT Kcv;           /* 0xC000706A : A/D transfer gain for iv      */
/*-----*/
/*    for over modulation                */
/*-----*/
    SHORT Kmod;          /* 0xC000706C : Over modulation gain/offset    */
    SHORT Kv;            /* 0xC000706E : AVR                          */
/*-----*/
    SHORT Kc;            /* 0xC0007070 : Current conversion Gain      */
    SHORT Ld;            /* 0xC0007072 : d axis Inductance            */
    SHORT Lq;            /* 0xC0007074 : q axis Inductance            */

```

---

---

```

SHORT Mag;          /* 0xC0007076 : Magnetic flux (Phi)          */
SHORT KdP;          /* 0xC0007078 : d axis propotion gain (PI control)        */
SHORT KqP;          /* 0xC000707A : q axis propotion gain (PI control)        */
SHORT KdI;          /* 0xC000707C : d axis integral time (gain)               */
SHORT KqI;          /* 0xC000707E : q axis integral time (gain)               */
SHORT VdLim;        /* 0xC0007080 : d axis integral limit                     */
SHORT VqLim;        /* 0xC0007082 : q axis integral limit                     */
SHORT KvIn;         /* 0xC0007084 : Voltage Compasation gain                  */
SHORT OnDelayLvl;   /* 0xC0007086 : On delay change level                     */
SHORT Tfil;         /* 0xC0007088 : Filter time constant                     */
SHORT Vmax;         /* 0xC000708A : voltage limit data (Vmax^2)               */
SHORT OnDelayComp;  /* 0xC000708C : On-delay compensation                     */
SHORT CtrlSw;       /* 0xC000708E : Control Flag                              */
USHORT CrFreq;      /* 0xC0007090 : carrier freq reference(change while BB)  */
SHORT CtrlSw2;      /* 0xC0007092 : Control Flag2                            */
SHORT Dummy0;       /* 0xC0007094 : for Alignment                             */
/*-----*/
/*      notch1      */
/*-----*/
SHORT Kf11;         /* 0xC0007096 :          */
SHORT Kf12;         /* 0xC0007098 :          */
SHORT Kf13;         /* 0xC000709A :          */
SHORT Kf14;         /* 0xC000709C :          */
/*-----*/
/*      notch2      */
/*-----*/
SHORT Kf21;         /* 0xC000709E :          */
SHORT Kf22;         /* 0xC00070A0 :          */
SHORT Kf23;         /* 0xC00070A2 :          */
SHORT Kf24;         /* 0xC00070A4 :          */
/*-----*/
/*      notch3      */
/*-----*/
SHORT Kf31;         /* 0xC00070A6 :          */
SHORT Kf32;         /* 0xC00070A8 :          */
SHORT Kf33;         /* 0xC00070AA :          */
SHORT Kf34;         /* 0xC00070AC :          */

```

---

---

```

/*-----*/
/*      notch4                      */
/*-----*/
SHORT Kf41;      /* 0xC00070AE :          */
SHORT Kf42;      /* 0xC00070B0 :          */
SHORT Kf43;      /* 0xC00070B2 :          */
SHORT Kf44;      /* 0xC00070B4 :          */
/*-----*/
/*      notch5                      */
/*-----*/
SHORT Kf51;      /* 0xC00070B6 :          */
SHORT Kf52;      /* 0xC00070B8 :          */
SHORT Kf53;      /* 0xC00070BA :          */
SHORT Kf54;      /* 0xC00070BC :          */
/*-----*/
/*      notch6                      */
/*-----*/
SHORT Kf61;      /* 0xC00070BE :          */
SHORT Kf62;      /* 0xC00070C0 :          */
SHORT Kf63;      /* 0xC00070C2 :          */
SHORT Kf64;      /* 0xC00070C4 :          */
/*-----*/
/*      LPF                        */
/*-----*/
SHORT TLpf;      /* 0xC00070C6 : LPF1 gain          */
SHORT TLpf2;     /* 0xC00070C8 : LPF2 gain          */
/*-----*/
SHORT MotResist; /* 0xC00070CA : Moter resistance(r1/I_BASE*V_BASE)          */
SHORT OnDelaySlope; /* 0xC00070CC : Dead-time comp. gain          */
SHORT L_dIdt;     /* 0xC00070CE :          */
SHORT FccRst;     /* 0xC00070D0 : FCCST Reset(for TEST)          */
SHORT Dummy;     /* 0xC00070D2 : for Alignment          */
} INTADP;

/*-----*/
/*      Current Loop Control Data          */
/*-----*/

```

---

---

```

typedef struct _INTADV {
    SHORT IuOffset;      /* 0xC00070D4 : A/D transfer offset for iu          */
    SHORT IvOffset;      /* 0xC00070D6 : A/D transfer offset for iv          */
    /*-----*/
    SHORT IuInData;      /* 0xC00070D8 : iu input data (from A/D)          */
    SHORT IvInData;      /* 0xC00070DA : iv input data (from A/D)          */
    SHORT IdInData;      /* 0xC00070DC : Id Input                          */
    SHORT IqInData;      /* 0xC00070DE : Iq Input                          */
    /*-----*/
    USHORT CrFreqW;      /* 0xC00070E0 : Carrier freq Now                  */
    /*-----*/
    SHORT IuOut;          /* 0xC00070E2 : Iu reference(for dead-time cmp)    */
    SHORT IvOut;          /* 0xC00070E4 : Iv reference(for dead-time cmp)    */
    /*-----*/
    SHORT V1;            /* 0xC00070E6 : V1( = SQRT(VdOut^2+VqOut^2) )      */
    SHORT Vcent;          /* 0xC00070E8 : Central voltage                    */
    SHORT Dummy1;         /* 0xC00070EA : for Alignment                      */
    /*-----*/
    /* for LPF                */
    /*-----*/
    DWREG IqOut1Lpf;      /* 0xC00070EC : LPF1 Output(Low)                  */
    DWREG IqOut2Lpf;      /* 0xC00070F0 : LPF2 Output(Low)                  */
    /*-----*/
    SHORT IqRef;          /* 0xC00070F4 : Iq Reference after limit          */
    SHORT TLimP;          /* 0xC00070F6 : +tlim                             */
    SHORT TLimM;          /* 0xC00070F8 : -tlim                             */
    /*-----*/
    /* for monitor            */
    /*-----*/
    SHORT IqMon;          /* 0xC00070FA : IQ monitor                        */
    /*-----*/
    SHORT IdDataP;        /* 0xC00070FC :                                */
    SHORT IqDataP;        /* 0xC00070FE :                                */
    SHORT KEangle;        /* 0xC0007100 : 機 械 角 → 電 気 角変換係数          */
    SHORT Dummy2;         /* 0xC0007102 : for Alignment                      */
    DWREG IdLfIl;         /* 0xC0007104 : L(di/dt)フ イ ルタ                */
    DWREG IqLfIl;         /* 0xC0007108 :                                */

```

---

```

/*-----*/
/*   for axis q monitor                               */
/*-----*/
SHORT IqDistIn;    /* 0xC000710C : q軸 外 乱 ト ルク 入力 */
SHORT IqDist;      /* 0xC000710E : q軸 外 乱 ト ルク */
SHORT IqMonFil;    /* 0xC0007110 : q軸 電 流 指 令 モ ニ タ (フ ィ ル タ 後) */
SHORT IqOfRef;     /* 0xC0007112 : q軸 電 流 指 令 ( 外 乱 ト ルク 加 算 後) */
/*-----*/
/*   for notch filter1                               */
/*-----*/
DWREG IqOut1L;     /* 0xC0007114 : filter1 output (Low) */
DWREG IqOut1PL;    /* 0xC0007118 : filter1 output 1delay (Low) */
DWREG IqOut1PPL;   /* 0xC000711C : filter1 output 2delay (Low) */
DWREG IqIn1PL;     /* 0xC0007120 : filter1 input 1delay (Low) */
DWREG IqIn1PPL;    /* 0xC0007124 : filter1 input 2delay (Low) */
SHORT IQ01;        /* 0xC0007128 : filter1 output (short type) */
SHORT Dummy3;      /* 0xC000712A : for Alignment */
DWREG IqOut1BufL;  /* 0xC000712C : filter1 output buffer (Low) */
/*-----*/
/*   for notch filter2                               */
/*-----*/
DWREG IqOut2L;     /* 0xC0007130 : filter2 output (Low) */
DWREG IqOut2PL;    /* 0xC0007134 : filter2 output 1delay (Low) */
DWREG IqOut2PPL;   /* 0xC0007138 : filter2 output 2delay (Low) */
DWREG IqIn2PL;     /* 0xC000713C : filter2 input 1delay (Low) */
DWREG IqIn2PPL;    /* 0xC0007140 : filter2 input 2delay (Low) */
SHORT IQ02;        /* 0xC0007144 : filter2 output (short type) */
SHORT Dummy4;      /* 0xC0007146 : for Alignment */
DWREG IqOut2BufL;  /* 0xC0007148 : filter2 output buffer (Low) */
/*-----*/
/*   for notch filter3                               */
/*-----*/
DWREG IqOut3L;     /* 0xC000714C : filter3 output (Low) */
DWREG IqOut3PL;    /* 0xC0007150 : filter3 output 1delay (Low) */
DWREG IqOut3PPL;   /* 0xC0007154 : filter3 output 2delay (Low) */
DWREG IqIn3PL;     /* 0xC0007158 : filter3 input 1delay (Low) */
DWREG IqIn3PPL;    /* 0xC000715C : filter3 input 2delay (Low) */

```

---

```

SHORT IqOut3;      /* 0xC0007160 : filter3 output(short type)          */
SHORT Dummy5;      /* 0xC0007162 : for Alignment                                         */
DWREG IqOut3BufL;   /* 0xC0007148 : filter3 output buffer(Low)                           */
/*-----*/
/*   for notch filter4                                           */
/*-----*/
DWREG IqOut4L;      /* 0xC0007164 : filter4 output(Low)                                   */
DWREG IqOut4PL;     /* 0xC0007168 : filter4 output 1delay(Low)                           */
DWREG IqOut4PPL;    /* 0xC000716C : filter4 output 2delay(Low)                           */
DWREG IqIn4PL;      /* 0xC0007170 : filter4 input 1delay(Low)                            */
DWREG IqIn4PPL;     /* 0xC0007174 : filter4 input 2delay(Low)                            */
SHORT IQ04;         /* 0xC0007178 : filter4 output(short type)                           */
SHORT Dummy6;       /* 0xC000717A : for Alignment                                         */
DWREG IqOut4BufL;   /* 0xC000717C : filter4 output buffer(Low)                           */
/*-----*/
/*   for notch filter5                                           */
/*-----*/
DWREG IqOut5L;      /* 0xC0007180 : filter5 output(Low)                                   */
DWREG IqOut5PL;     /* 0xC0007184 : filter5 output 1delay(Low)                           */
DWREG IqOut5PPL;    /* 0xC0007188 : filter5 output 2delay(Low)                           */
DWREG IqIn5PL;      /* 0xC000718C : filter5 input 1delay(Low)                            */
DWREG IqIn5PPL;     /* 0xC0007190 : filter5 input 2delay(Low)                            */
SHORT IQ05;         /* 0xC0007194 : filter5 output(short type)                           */
SHORT Dummy7;       /* 0xC0007196 : for Alignment                                         */
DWREG IqOut5BufL;   /* 0xC0007198 : filter5 output buffer(Low)                           */
/*-----*/
/*   for notch filter6                                           */
/*-----*/
DWREG IqOut6L;      /* 0xC000719C : filter6 output(Low)                                   */
DWREG IqOut6PL;     /* 0xC00071A0 : filter6 output 1delay(Low)                           */
DWREG IqOut6PPL;    /* 0xC00071A4 : filter6 output 2delay(Low)                           */
DWREG IqIn6PL;      /* 0xC00071A8 : filter6 input 1delay(Low)                            */
DWREG IqIn6PPL;     /* 0xC00071AC : filter6 input 2delay(Low)                            */
SHORT IqOut6;       /* 0xC00071B0 : filter6 output(short type)                           */
SHORT Dummy8;       /* 0xC00071B2 : for Alignment                                         */
DWREG IqOut6BufL;   /* 0xC00071B4 : filter6 output buffer(Low)                           */
} INTADV;

```

---

---

```

/*-----*/
/*   for Current Observer                               */
/*-----*/
typedef struct _DOBSP {
    SHORT TsPerL;      /* 0xC00071B8 : Ts/L ( オ ブ ザ ー バ 用 パラメータ ) */
    SHORT Gobs;        /* 0xC00071BA : g ( オ ブ ザ ー バの極 ) */
    SHORT RLTS;        /* 0xC00071BC : 1-R・Ts/L ( オ ブ ザ ー バ 用 パラメータ ) */
    SHORT FilObsGain;   /* 0xC00071BE : フ ィ ル タ ゲイン */
} DOBSP;

/*-----*/
/*   for Current Observer                               */
/*-----*/
typedef struct _DOBSV {
    DWREG Lpfilq;      /* 0xC00071C0 : q軸 ロ ー パ ス フ ィ ル タ 変数 */
    DWREG Hpfilq;      /* 0xC00071C4 : q軸 ハ イ パ ス フ ィ ル タ 変数 */
    SHORT IqObsOut;     /* 0xC00071C8 : q軸 オ ブ ザ ー バ 出力 */
    SHORT IqObsFreq;    /* 0xC00071CA : q軸 振 動 成分 */
    DWREG LpfilD;      /* 0xC00071CC : d軸 ロ ー パ ス フ ィ ル タ 変数 */
    DWREG HpfilD;      /* 0xC00071D0 : d軸 ハ イ パ ス フ ィ ル タ 変数 */
    SHORT IdObsOut;     /* 0xC00071D4 : d軸 オ ブ ザ ー バ 出力 */
    SHORT IdObsFreq;    /* 0xC00071D6 : d軸 振 動 成分 */
    SHORT DmpGain;      /* 0xC00071D8 : 電 流 オ ブ ザ ー バ ダ ン ピ ン グ ゲイン */
    SHORT Dummy;        /* 0xC00071DA : for Alignment */
} DOBSV;

/*-----*/
/*   for Phase Interpolate                               */
/*-----*/
typedef struct _PHASEV {
    SHORT PHAL;        /* 0xC00071DC : Phase(Low) */
    SHORT PhaseH;      /* 0xC00071DE : Phase(High) */
    SHORT PhaseIp;     /* 0xC00071E0 : 位 相 補 間 量 */
    SHORT PhaseIpF;    /* 0xC00071E2 : 位 相 補 間 フ ラ グ (0 : し な い 、 1 : する) */
}

```

---



---

```

SHORT PhaseIpIn;      /* 0xC00071E4 : 位 相 補 間 量 (CPU→ A S I C 受 け 渡 し 用)          */
SHORT PhaseIpFin;     /* 0xC00071E6 : 位 相 補 間 フ ラ グ (C P U → A S I C 受 け 渡 し 用)          */
} PHASEV;

/*-----*/
/*      for Encoder IF                                */
/*-----*/
typedef struct _ENCIFV {
SHORT AmpType;        /* 0xC00071E8 : Rotary Type or Linear Type          */
/*-----*/
SHORT Dummy;          /* 0xC00071EA : for Alignment                        */
DWREG RcvPosX0;        /* 0xC00071EC : Motor Encoder Position(受 信 位 置)          */
DWREG RcvPosX1;        /* 0xC00071F0 : Motor Encoder Position(受 信 位 置)          */
DWREG RcvPosX2;        /* 0xC00071F4 : Motor Encoder Position(受 信 位 置)          */
/*-----*/
SHORT MotPosSftR;      /* 0xC00071F8 : Shift Data for Motor Position Cal.          */
SHORT MotPosSftX;      /* 0xC00071FA : Shift Data for Motor Position Cal.          */
SHORT MotPosSign;      /* 0xC00071FC : Sign Data for Motor Position Cal.          */
SHORT DivOutSft;       /* 0xC00071FE : 分 周 出 力 シ フ ト 数          */
DWREG AccErrLv;        /* 0xC0007200 : Motor Encoder Acc. Error Check Low          */
/*-----*/
DWREG DivPls;          /* 0xC0007204 : 分 周 出 力 パ ル ス          */
/*-----*/
DWREG DivOutGain;      /* 0xC0007208 : 分 周 出 力 ゲ イ ン (リニア)          */
DWREG DivPos;          /* 0xC000720C : 分 周 出 力 パ ル ス (リニア)          */
DWREG DivPlsRem;       /* 0xC0007210 : 分 周 出 力 パ ル ス 余 り (リニア)          */
/*-----*/
SHORT SPGFail;         /* 0xC0007214 : S-PG受 信 失 敗 許 容 回 数          */
/*-----*/
SHORT BitIprm;         /* 0xC0007216 : Bit Initial Parameter          */
SHORT BitData;         /* 0xC0007218 : Bit Data          */
/*-----*/
SHORT IncPlsReq;       /* 0xC000721A : PA0 IncrePulse Output Request          */
SHORT PA0SeqCmd;       /* 0xC000721C : PA0 Serial Output Sequence          */
SHORT Pls0SetCmd;      /* 0xC000721E : パ ル ス 出 力 回 路 初 期 化 要 求          */
} ENCIFV;

```

---

```

/*-----*/
/*   分 周 パ ル ス 出力関連 : HostCPU  - - >   A s i c 定周期転送用           */
/*-----*/
typedef struct _DIVPLSV {
    SHORT IncPlsReqIn;    /* 0xC0007220 : PA0 IncrePulse Output Request      */
    SHORT PA0SeqCmdIn;    /* 0xC0007222 : PA0 Serial Output Sequence          */
    SHORT Pls0SetCmdIn;   /* 0xC0007224 : パ ル ス 出 力 回 路初期化要求      */
/*-----*/
    SHORT DivSetIn;       /* 0xC0007226 : 分 周 機 能 設定入力                */
    SHORT PoSet1In;       /* 0xC0007228 : パ ル ス 変 換 原点補正1            */
    SHORT PoSet2In;       /* 0xC000722A : パ ル ス 変 換 原点補正2            */
/*-----*/
    DWREG IncInitPlsIn;   /* 0xC000722C : 初 期 イ ン ク レパルス            */
    DWREG IncInitRemIn;   /* 0xC0007230 : 初 期 イ ン ク レパルス余り        */
/*-----*/
    SHORT AccCntClrReq;   /* 0xC0007234 : 加 速 度 チ ェ ッ ク 開 始 カウ ン トクリア要求 */
/*-----*/
    SHORT Dummy;         /* 0xC0007236 : for Alignment                      */
/*-----*/
/*   Register for Library           */
/*-----*/
    DWREG Argu0;          /* 0xC0007238 : Argument0                        */
    DWREG Argu1;          /* 0xC000723C : Argument1                        */
    SHORT Argu2;          /* 0xC0007240 : Argument2                        */
    SHORT Argu2H;         /* 0xC0007242 : Argument2 High Word              */
    DWREG Ret0;           /* 0xC0007244 : Return Register                  */
    DWREG Kx;             /* 0xC0007248 : kx                               */
    DWREG Sx;             /* 0xC000724C : sx                               */
    DWREG Iu0;            /* 0xC0007250 : iu[0]                            */
    SHORT Iu1;            /* 0xC0007254 : iu[1]                            */
    SHORT Iu1H;           /* 0xC0007256 : iu[1] High Word                  */
} DIVPLSV;

/*-----*/
/*   for Weaken Field           */
/*-----*/
typedef struct _WEAKFV {

```

---

```

SHORT WfKpVLIn;      /* 0xC0007258 : 電 圧F B 比 例 ゲ イン(下位1 6b it) */
SHORT WfKpVHIn;      /* 0xC000725A : 電 圧F B 比 例 ゲ イン(上位1 6b it) */
SHORT WfKiVLIn;      /* 0xC000725C : 電 圧F B 積 分 ゲ イン(下位1 6b it) */
SHORT WfKiVHIn;      /* 0xC000725E : 電 圧F B 積 分 ゲ イン(上位1 6b it) */
SHORT WfVlMaxIn;      /* 0xC0007260 : 電 圧 指 令 制 限レベル */
SHORT WfIdRefLimIn;   /* 0xC0007262 : d軸 電 流 指 令 リミット */
/*-----*/
SHORT WfIntegLim;     /* 0xC0007264 : 電 圧F B 積 分 リミット */
/*-----*/
SHORT IdOut;          /* 0xC0007266 : Id reference */
SHORT IqOut;          /* 0xC0007268 : Iq reference */
SHORT Vel;            /* 0xC000726A : Velocity (omega) */
/*-----*/
DWREG WfKpV;          /* 0xC000726C : 電 圧F B 比 例 ゲ イン(下位1 6b it) */
DWREG WfKiV;          /* 0xC0007270 : 電 圧F B 積 分 ゲ イン(下位1 6b it) */
SHORT WfVlMax;        /* 0xC0007274 : 電 圧 指 令 制 限レベル */
SHORT WfIdRefLim;     /* 0xC0007276 : d軸 電 流 指 令 リミット */
SHORT WfVqMax;        /* 0xC0007278 : q軸 電 圧 指 令 リミット */
/*-----*/
SHORT Dummy;          /* 0xC000727A : for Alignment */
DWREG WfIntgl;        /* 0xC000727C : 電 圧F B 積 分値 */
SHORT WfVdRef;        /* 0xC0007280 : d軸 電 圧 指 令 */
SHORT WfVqRef;        /* 0xC0007282 : q軸 電 圧 指 令 */
} WEAKFV;

/*-----*/
/* for make PWM Carrer Counter */
/*-----*/
typedef struct _PWMV {
SHORT PwmCntT2;      /* 0xC0007220 : PWMキ ャ リ ア カウンタT2 */
SHORT PwmCntT1;      /* 0xC0007222 : PWMキ ャ リ ア カウンタT1 */
SHORT PwmCntT0;      /* 0xC0007224 : PWMキ ャ リ ア カウンタT0 */
SHORT Dummy;        /* 0xC0007226 : for Alignment */
} PWMV;

/*-----*/
/* Version define register */

```

---

---

```

/*-----*/
typedef struct _VER_INFO {
    SHORT MswVer;      /* 0xC0007228 : ソフトバージョン情報 */
    SHORT TstVer;      /* 0xC000722A : テストバージョン情報 */
    SHORT YspVer;      /* 0xC000722C : Y仕様バージョン情報 */
    SHORT Dummy;       /* 0xC000722E : for Alignment */
} VER_INFO;

/*-----*/
/* Common work register */
/*-----*/
typedef struct _COMWORKS {
    SHORT WREG82;      /* 0xC0007290 : Work Register 82 */
    SHORT WREG83;      /* 0xC0007292 : Work Register 83 */
    SHORT WREG84;      /* 0xC0007294 : Work Register 84 */
    SHORT WREG85;      /* 0xC0007296 : Work Register 85 */
    SHORT WREG86;      /* 0xC0007298 : Work Register 86 */
    SHORT WREG87;      /* 0xC000729A : Work Register 87 */
    SHORT WREG88;      /* 0xC000729C : Work Register 88 */
    SHORT WREG89;      /* 0xC000729E : Work Register 89 */
    SHORT WREG95;      /* 0xC00072A0 : Work Register 95 */
    SHORT WREG100;     /* 0xC00072A2 : Work Register 100 */
    SHORT WREG101;     /* 0xC00072A4 : Work Register 101 */
    SHORT WREG102;     /* 0xC00072A6 : Work Register 102 */
    SHORT WREG103;     /* 0xC00072A8 : Work Register 103 */
    SHORT WREG104;     /* 0xC00072AA : Work Register 104 */
    SHORT WREG109;     /* 0xC00072AC : Work Register 109 */
    SHORT Dummy;       /* 0xC00072AE : for Alignment */
} COMWORKS;

typedef struct _HOST_WOKS {
    DWREG HTMP0;       /* 0xC00072B0 : HOST Interrupt Temp0 */
    DWREG HTMP2;       /* 0xC00072B4 : HOST Interrupt Temp2 */
    DWREG HTMP4;       /* 0xC00072B8 : HOST Interrupt Temp4 */
    DWREG HTMP6;       /* 0xC00072BC : HOST Interrupt Temp6 */
    SHORT HTMP7;       /* 0xC00072C0 : HOST Interrupt Temp7 */
    SHORT HTMP8;       /* 0xC00072C2 : HOST Interrupt Temp8 */
}

```

---

---

```

} HOST_WOKS;

typedef struct _ROUND_WORKS {
    SHORT WREG140;    /* 0xC00072C4 : Work Register 140 */
    SHORT WREG141;    /* 0xC00072C6 : Work Register 141 */
    SHORT WREG142;    /* 0xC00072C8 : Work Register 142 */
    SHORT WREG143;    /* 0xC00072CA : Work Register 143 */
    SHORT WREG144;    /* 0xC00072CC : Work Register 144 */
    SHORT WREG145;    /* 0xC00072CE : Work Register 145 */
    SHORT WREG146;    /* 0xC00072D0 : Work Register 146 */
    SHORT WREG147;    /* 0xC00072D2 : Work Register 147 */
    SHORT WREG148;    /* 0xC00072D4 : Work Register 148 */
    SHORT WREG149;    /* 0xC00072D6 : Work Register 149 */
} ROUND_WORKS;

typedef struct _AD_WORKS {
    DWREG TMP0;    /* 0xC00072D8 : AD Interrupt Temp0 */
    DWREG TMP2;    /* 0xC00072DC : AD Interrupt Temp2 */
    DWREG TMP4;    /* 0xC00072E0 : AD Interrupt Temp4 */
    DWREG TMP6;    /* 0xC00072E4 : AD Interrupt Temp6 */
    DWREG TMP8;    /* 0xC00072E8 : AD Interrupt Temp8 */
} AD_WORKS;

/*-----*/
/*  初  期  化  処  理  用  ワ ー  ク                               */
/*-----*/
typedef struct _INITWK {
    SHORT IN_WK0;    /* 0xC00072EC : 初  期  化  処理用 Work0 Lo */
    SHORT IN_WK0H;    /* 0xC00072EE : 初  期  化  処理用 Work0 Hi */
    SHORT IN_WK1;    /* 0xC00072F0 : 初  期  化  処理用 Work1 Lo */
    SHORT IN_WK1H;    /* 0xC00072F2 : 初  期  化  処理用 Work1 Hi */
} INITWK;

/*****/

```

---

---

```

/*  Axis Control Structure definition  */
/*****
typedef struct {
    AXIS_INFO  AxisInfo; /* 軸 情 報 変 数  */
    ACRV      AcrV; /* 電 流 制 御 用 変 数  */
    STSFLG    StsFlg; /* ステータスフラグ */
    VCMPIV    VcmpV; /* 電 圧 補 償 演 算 用 変 数  */
    SINTBL     SinTbl; /* sinテ ー ブル */
    ADSTOP     AdStop; /* 電 流 制 御 停 止 エ ラ ー 検 出 用 構 造 体  */
    ADINV      AdinV; /* ホ ス ト C P U 受 け 渡 し 変 数  */
    INTADP     IntAdP; /* 電 流 割 込 み パ ラ メ ー タ  */
    INTADV     IntAdV; /* 電 流 割 込 み 変 数  */
    DOBSP      DobsP; /* d軸 オ ブ ザ ー バ 用 パ ラ メ ー タ  */
    DOBSV      DobsV; /* d軸 オ ブ ザ ー バ 用 変 数  */
    PHASEV     PhaseV; /* 位 相 演 算 用 変 数  */
    ENCIFV     EncIfV; /* エ ン コ ー ダ イ ン タ フ ェ ー ス 変 数  */
    DIVPLSV    DivPlsV; /* 分 周 パ ル ス 用 変 数  */
    WEAKFV     WeakFV; /* 弱 め 界 磁 用 変 数  */
    PWMV       PwmV; /* PWMキ ャ リ ア カ ウ ン タ 格 納 変 数  */
    USHORT     CtrlStsIn; /* μ プ ロ グ ラ ム 制 御 ス テ ー タ ス (CPU⇒μ プ ロ グ ラ ム) */
    USHORT     CtrlStsOut; /* μ プ ロ グ ラ ム 制 御 ス テ ー タ ス (μ プ ロ グ ラ ム⇒CPU) */
    SHORT      Dummy[305]; /* Dummy buffer for Axis offset */
    SVIP_READ_REG *SvIpRegR; /* Servo IP H/W Read Register Address */
    SVIP_WRITE_REG *SvIpRegW; /* Servo IP H/W Write Register Address */
} MICRO_AXIS_HANDLE;

extern MICRO_AXIS_HANDLE AxisHdl[2];
extern LONG AxisNum;
extern SHORT ZERO;
extern SHORT ONE;
extern LONG True;
extern LONG False;
extern VER_INFO VerInfo;

#define ZEROR 0x00000000 /* 2013.05.06 tanaka21 コ ー ド 整 理 (マクロ化) <022> */
#define ONER 0x00000001 /* 2013.05.06 tanaka21 コ ー ド 整 理 (マクロ化) <022> */
#define NONER 0xffffffff /* 2013.05.06 tanaka21 コ ー ド 整 理 (マクロ化) <022> */

```

---

---

/\*-----\*/

/\*\*\*\*\* end of file \*\*\*\*\*/