

```

/*****
/*
/*
/*   MprgSgdv.inc : S   G   D   V   マ イ ク ロ プ ロ グ ラ ム           */
/*
/*
/*****
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/***** Copyright (C) Yaskawa Electric Corporation *****/
/*
/*   Rev.0.00 : 2012.08.06 Y.Tanaka   ・ JL-086 向 け 電 流 制 御 C 言 語 化 用 にバ ー ジョン取り直し   */
/*   Rev.0.01 : 2012.08.17 Y.Tanaka   ・ 構 造 体 、 ロ ー カル変数見直し           */
/*   Rev.0.02 : 2012.08.20 Y.Tanaka   ・ 構 造 体 、 ロ ー カル変数見直し           */
/*
/*****

/*****
/*
/*   マ イ ク ロ 演 算 部 ソ フ トバ ー ジョン 定義           <V720>   */
/*
/*
/*****
//#pragma AsicReg begin
#define MSW_VER    0x0002    /* ADR:0x0002: ソ フ ト バ ー ジ ョ ン(製品用)           */
#define TST_VER    0x0000    /* ADR:0x0000: テ ス ト バ ー ジ ョ ン(評価用)           */
#define YSP_VER    0x0000    /* ADR:0x0000: Y 仕 様 バ ー ジ ョ ン(製品用)           */

```

```

/*****
/*
/*      Function Prototype
/*
/*****
#ifdef ASIP_CC      /* JL-086コンパイラ用
void main( void ); /* スタートアップ
#elif defined(WIN32) /* VCシミュレーション動作
void MpStart( void ); /* スタートアップ
#endif
void MpIntHost( void ); /* ホスト 割込み処理
void MpIntAD( void ); /* 電 流 制 御 割込み
void MpIntEnc( void ); /* エンコーダ割込み
void MpUPDATE_DIVPOS( void ); /* 分 周 位 置更新
void MpDataClear( void ); /* マイクロ用 データクリア
USHORT MpSQRT( ULONG src ); /* 平 方 根 演 算 処理(整数)
void MpOVMMODK( void ); /* オ ー バ モ ジ ュ ール処理?
void MpMlibPfbkxremNolim( void ); /* 余 り 付き F B 演算処理

/*****
/*
/*      H/W Register
/*
/*****
/*-----*/
/*      READ register
/*-----*/
SHORT IuAD; /* ADR:0x07d0: AD data U
SHORT IvAD; /* ADR:0x07d1: AD data V
// --; /* ADR:0x07d2:
/*-----*/
SHORT IFLAG; /* ADR:0x07d3: Flag status
// bit.0 : Z(zero)
// bit.1 : S(sign'-')
// bit.2 : C(carry)

```

---

```

        // bit.3 : B(bollow)
        // bit.4 : LB(limit)
        // bit.5 : ZD(zero div)
        // bit.6 : DOVF(over-flow div)
        // bit.8 : ILS0(interrupt0)
        // bit.9 : ILS1(interrupt1)
        // bit.A : ILS2(interrupt2)

/*-----*/
DWREG MacL;          /* ADR:0x07d4: MPLY 0-15 bit          */
#define MACCL MacL.l  /* ADR:0x07d4: MPLY 0-31 bit          */
#define MACCLL MacL.s[0] /* ADR:0x07d4: MPLY 0-15 bit          */
#define MACCLH MacL.s[1] /* ADR:0x07d4: MPLY 16-31 bit          */
/*-----*/

DWREG MacH;          /* ADR:0x07d6: MPLY 32-47 bit          */
#define MACCH MacH.l  /* ADR:0x07d6: MPLY 32-63 bit          */
#define MACCHL MacH.s[0] /* ADR:0x07d6: MPLY 32-47 bit          */
#define MACCHH MacH.s[1] /* ADR:0x07d6: MPLY 48-63 bit          */
/*-----*/

SHORT DivRem;        /* ADR:0x07d8: Divide modulo          */
SHORT HSUR0;          /* ADR:0x07d9: ホ ス ト 指 令 ポー ト 0          */
SHORT HSUR1;          /* ADR:0x07da: ホ ス ト 指 令 ポー ト 1          */
SHORT HSUR2;          /* ADR:0x07db: ホ ス ト 指 令 ポー ト 2          */
SHORT HSUR3;          /* ADR:0x07dc: ホ ス ト 指 令 ポー ト 3          */
/*-----*/

SHORT CTSTR;          /* ADR:0x07dd: controll status          */
        // bit.0 : '1' 固 定
        // bit.1 : CTRD1
        // bit.2 : CTRD2
        // bit.3 : CTRD3(RLOCK)
        // bit.4 : CTRD4(BB)
        // bit.5 : CTRD5
        // bit.6 : CTRD6
        // bit.7 : CTRD7

/*-----*/
SHORT INT1SET;        /* ADR:0x07de: INT1 status read clr          */
        // bit.0 : EADINTAC
        // bit.1 : MPINTAC

```

---

---

```

        // bit.2 : CNVSTAC
        // bit.3 : INTOAC
        // bit.4 : CMDINTAC
        // bit.5 : SPGOINTAC
        // bit.6 : SPG1INTAC
        // bit.7 : URTINTAC
/*-----*/
SHORT FLTSTAT;          /* ADR:0x07df: fault status (FLTSTAT)          */
        // bit.0 : HBLK0(Sync-Serial bit0)
        // bit.1 : HBLK1(Sync-Serial bit1)
        // bit.2 : HBLK2(Sync-Serial bit2)
        // bit.3 : '0' 固定
        // bit.4 : OV DAT(OV)
        // bit.5 : OC DAT(OC)
        // bit.6 : OVLG(Digital filter fault:0)
        // bit.7 : DBON1LT
        // bit.8 : DBON2LT
        // bit.A : BB(Base-Block)
        // bit.B : GUP2(Give-up host:0)
        // bit.C : OVLGCH1(fault:0)
        // bit.D : OVLGCH2(fault:0)
        // bit.E : OVLGCH3(fault:0)
        // bit.F : CRUP(carrier counter status:up=1)
/*-----*/
SHORT FBCSET;           /* ADR:0x07e0: counter status          */
        // bit.0:PZD(test)
        // bit.1 : DADAT(division out A)
        // bit.2 : DBDAT(division out B)
        // bit.3 : DCDAT(division out C)
        // bit.4 : PADAT(PA)
        // bit.5 : PBDAT(PB)
        // bit.6 : PC DAT(PC)
        // bit.7 : ZRE
        // bit.8 : CADAT(CA input)
        // bit.9 : CBDAT(CB input)
        // bit.A : CCDAT(CC input)
        // bit.B : CLE(CLM latch end)

```

---

---

```

        // bit.C : PUI(U)
        // bit.D : PVI(V)
        // bit.E : PWI(W)
        // bit.F : UE
/*-----*/
SHORT FCCST;        /* ADR:0x07e1: SDM status */
        // bit.0 : SDM1TER(decimation1 over)
        // bit.1 : SDM2TER(decimation2 over)
        // bit.2 : SDM3TER(decimation3 over)
        // bit.3 : SDM1TER(decimation1 mclk miss)
        // bit.4 : SDM2TER(decimation2 mclk miss)
        // bit.5 : SDM3TER(decimation3 mclk miss)
        // bit.7 : Encoder AD Latch Complete flag)
        // bit.8 : RXFLG0(Serial-Enc0 receive flag)
        // bit.9 : RXFLG1(Serial-Enc1 receive flag)
/*-----*/
SHORT CRUDP;        /* ADR:0x07e2: Y-Tcounter */
        // bit.0-2 : for generic A/D
        // bit.3 : '1' 固定
        // bit.4-6 : for decimation1
        // bit.8-A : for decimation2
/*-----*/
// --;        /* ADR:0x07e3: */
SHORT OPCCLM;        /* ADR:0x07e4: OP counter CLM latch */
// --;        /* ADR:0x07e5: */
// --;        /* ADR:0x07e6: */
SHORT OPCLT;        /* ADR:0x07e7: OP counter load latch */
// --;        /* ADR:0x07e8: */
// --;        /* ADR:0x07e9: */
SHORT SRPG1RD2;        /* ADR:0x07ea: Serial-PG1 receive word2 */
SHORT SRPG1RD3;        /* ADR:0x07eb: Serial-PG1 receive word3 */
SHORT SRPG1RD4;        /* ADR:0x07ec: Serial-PG1 receive word4 */
SHORT SRPG1RD5;        /* ADR:0x07ed: Serial-PG1 receive word5 */
SHORT SRPG1RD6;        /* ADR:0x07ee: Serial-PG1 receive word6 */
SHORT SRPG1RD7;        /* ADR:0x07ef: Serial-PG1 receive word7 */
/*-----*/
DWREG SftMacL;        /* ADR:0x07f0: BRL shifter 0-15 bit */

```

---

---

```

DWREG SftMacH;          /* ADR:0x07f2: BRL shifter 32-47 bit          */
SHORT CAD0D;            /* ADR:0x00f4: Operation AD0 data          */
SHORT CAD1D;            /* ADR:0x00f5: Operation AD1 data Low      */
SHORT CAD1H;            /* ADR:0x00f6: Operation AD1 data high     */
//  --;                /* ADR:0x00f7:                                */
/*-----*/
SHORT ILR;              /* ADR:0x07f8: interrupt level read        */
// bit.0-2 : interrupt0
// bit.4-6 : interrupt1
// bit.8-A : interrupt2
// bit.C-E : interrupt3
/*-----*/
//  --;                /* ADR:0x07f9:                                */
SHORT SRPGORD2;         /* ADR:0x07fa: serial PG0 Rx word 2        */
SHORT SRPGORD3;         /* ADR:0x07fb: serial PG0 Rx word 3        */
SHORT SRPGORD4;         /* ADR:0x07fc: serial PG0 Rx word 4        */
SHORT SRPGORD5;         /* ADR:0x07fd: serial PG0 Rx word 5        */
SHORT SRPGORD6;         /* ADR:0x07fe: serial PG0 Rx word 6        */
SHORT SRPGORD7;         /* ADR:0x07ff: serial PG0 Rx word 7        */
//;
/*-----*/
/*  WRITE register          */
/*-----*/
SHORT UPD0;             /* ADR:0x07c0: 拡張IO for debug            <V720> */
/*-----*/
SHORT OUTPT;            /* ADR:0x07d0: INT2 port                    */
SHORT WDT1L;            /* ADR:0x07d1: WDT trigger port(WDT1L)      */
/*-----*/
SHORT BBSET;            /* ADR:0x07d2: soft BB set                  */
// bit.2 : soft BB
// bit.8 : EADINTEN(エンコーダAD変換完了割込みイネーブル)
// bit.9 : MPINTEN(M-Port割込みイネーブル)
// bit.A : CNVSTEN(enable CONVSTL int)
// bit.B : INTOEN
// bit.C : CMDINTEN
// bit.D : SPG0INTEN(シリアルエンコーダ0/C相エッジ割込みイネーブル)
// bit.E : SPG1INTEN(シリアルエンコーダ1割込みイネーブル)

```

---

---

```

// bit.F : UTINTEN(非同期シリアル送受信完割込みイネーブル)
/*-----*/
SHORT CRST;      /* ADR:0x07d3: Carrer start */
                // bit.0 : CRSTL(PWMキャリア開始指令)
                // bit.1 : CC40M(キャリアカウンタクロック設定 0:8 0MHz 1:40MHz)
SHORT SOD0L;     /* ADR:0x07d4: Sync-Serial0 Low */
SHORT SOD0H;     /* ADR:0x07d5: Sync-Serial0 High */
SHORT SOD1L;     /* ADR:0x07d6: Sync-Serial1 Low */
SHORT SOD1H;     /* ADR:0x07d7: Sync-Serial1 High */
/*-----*/
SHORT SDMECLR;   /* ADR:0x07d8: SDM status clr */
                // bit.0 : Decimation1 error clear
                // bit.1 : Decimation2 error clear
                // bit.2 : Decimation3 error clear
                // bit.3 : Decimation1-3 MCLK error clear
/*-----*/
SHORT ADSYNC;    /* ADR:0x07d9: AD sync */
// --;          /* ADR:0x07da: */
/*-----*/
SHORT PWMOS;     /* ADR:0x07db: PWM out sellect */
                // bit.0 : PWMSEL(0:Triangle,1:Space-Vec)
                // bit.3 : LV2L(0:2level,1:3level)
                // bit.4 : MBCMPH(0:Normal,1:Nx<--X2)
                // bit.5 : LDTYPE(0:Carrier,1:T0(servo))
                // bit.6 : FLDTYPE(0:Bottom,1:Top&Bottom)
                // bit.7 : SAWL(0:SAW type,1:Normal)
                // bit.8 : SAWDWL(0:Down,1:Up)
/*-----*/
SHORT CrSet1;    /* ADR:0x07dc: Carrer setting */
                // bit.0-2 : T&B counter clear set
/*-----*/
SHORT CTSTW;     /* ADR:0x07dd: controll status */
                // bit.0 : STD0
                // bit.1 : STD1
                // bit.2 : STD2
                // bit.3 : STD3(D/QLIM)
                // bit.4 : STD4

```

---

---

```

        // bit.5 : STD5(TLIM)
        // bit.6 : STD6
        // bit.7 : STD7

/*-----*/
SHORT ADT0;      /* ADR:0x07de: carrer 1/3 point set */
SHORT CRFRQ;     /* ADR:0x07df: Carrer frequency */
SHORT PwmP0;     /* ADR:0x07e0: PWM pattern 1 0 */
SHORT PwmP2;     /* ADR:0x07e1: PWM pattern 3 2 */
SHORT PwmP4;     /* ADR:0x07e2: PWM pattern 5 4 */
SHORT PwmP6;     /* ADR:0x07e3: PWM pattern 7 6 */
SHORT PwmP8;     /* ADR:0x07e4: PWM pattern 9 8 */
SHORT PwmP10;    /* ADR:0x07e5: PWM pattern 11 10 */
SHORT PwmP12;    /* ADR:0x07e6: PWM pattern 12 */
SHORT PwmT0;     /* ADR:0x07e7: PWM timer 0 */
SHORT PwmT1;     /* ADR:0x07e8: PWM timer 1 */
SHORT PwmT2;     /* ADR:0x07e9: PWM timer 2 */
SHORT PwmT3;     /* ADR:0x07ea: PWM timer 3 */
SHORT PwmT4;     /* ADR:0x07eb: PWM timer 4 */
SHORT PwmT5;     /* ADR:0x07ec: PWM timer 5 */
SHORT PwmT6;     /* ADR:0x07ed: PWM timer 6 */
SHORT PwmT7;     /* ADR:0x07ee: PWM timer 7 */
SHORT PwmT8;     /* ADR:0x07ef: PWM timer 8 */
SHORT PwmT9;     /* ADR:0x07f0: PWM timer 9 */
SHORT PwmT10;    /* ADR:0x07f1: PWM timer 10 */
SHORT PwmT11;    /* ADR:0x07f2: PWM timer 11 */
// --;          /* ADR:0x07f3: */
USHORT ISA0;     /* ADR:0x07f4: interrupt 0 address */
USHORT ISA1;     /* ADR:0x07f5: interrupt 1 address */
USHORT ISA2;     /* ADR:0x07f6: interrupt 2 address */
USHORT ISA3;     /* ADR:0x07f7: interrupt 3 address */
SHORT INTLVWR;   /* ADR:0x07f8: interrupt level write */
/*-----*/
SHORT DivSet;    /* ADR:0x07f9: counter divide set */
        // bit.0 : PCMASK
        // bit.A-B : division C out select
        // (00:normal,01:div
        // 10:normal,11:div(00|01))

```

---



---

```

/*-----*/
SHORT PCVS0;      /* ADR:0x07fa: PWM-pulse convert set0      */
SHORT PCVS1;      /* ADR:0x07fb: PWM-pulse convert set1      */
SHORT PCVS2;      /* ADR:0x07fc: PWM-pulse convert set2      */
SHORT EIX;        /* ADR:0x07fd: interrupt enable            */
SHORT DIX;        /* ADR:0x07fe: interrupt disable           */
//    -;          /* ADR:0x07ff:                             */
/*-----*/
/*    Flags              */
/*-----*/

/*****
/*
/*    定 数 マ ク ロ 定 義
/*
/*
/*****
#define ZEROR 0      /* ADR:0x0000: (WREGBASE+000) ; Zero register(Low)      */
#define ONER 1      /* ADR:0x01ec: one value register      ADDR = 080073D8 */
#define NONER -1    /* ADR:0x01ee: -1 value register      ADDR = 080073DC */

/*-----*/
/*    AMPTYPE : 回 転 型 / リ ニ ア 型 選 択          <V720> */
/*-----*/
//#pragma AsicReg begin
#define ROTARY 0x0000 /* 0 : 回 転 型          */
#define LINEAR 0x0001 /* 1 : リ ニ ア 型          */

/*-----*/
/*    PAOSEQCMD : PaoSeqMcmd Interface Definition          <V720> */
/*-----*/
#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) <V720> */

```

---

```

/*-----*/
#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 */

/*-----*/
/* BITIPRM <V720> */
/*-----*/
#define UPGDIVOUT 0x0001 /* bit0 : マ イ ク ロ 分 周 機能使用選択 */
// equ 0x0002 /* bit1 : 空 き */
// equ 0x0004 /* bit2 : 空 き */
// equ 0x0008 /* bit3 : 空 き */
// equ 0x0010 /* bit4 : 空 き */
// equ 0x0020 /* bit5 : 空 き */
// equ 0x0040 /* bit6 : 空 き */
// equ 0x0080 /* bit7 : 空 き */
// equ 0x0100 /* bit8 : 空 き */
// equ 0x0200 /* bit9 : 空 き */
// equ 0x0400 /* bitA : 空 き */
// equ 0x0800 /* bitB : 空 き */
// equ 0x1000 /* bitC : 空 き */
// equ 0x2000 /* bitD : 空 き */
// equ 0x4000 /* bitE : 空 き */
// equ 0x8000 /* bitF : 空 き */

/*-----*/
/* BITDAT <V720> */
/*-----*/
#define SERXERR 0x0001 /* bit0 : Senc Receive Error */
#define ACCCHKENA 0x0002 /* bit1 : Encoder Acc. Check Enable Flag */
// equ 0x0004 /* bit2 : 空 き */
// equ 0x0008 /* bit3 : 空 き */
// equ 0x0010 /* bit4 : 空 き */
// equ 0x0020 /* bit5 : 空 き */
// equ 0x0040 /* bit6 : 空 き */
// equ 0x0080 /* bit7 : 空 き */

```

---

```

//      equ      0x0100    /* bit8 : 空      き          */
//      equ      0x0200    /* bit9 : 空      き          */
//      equ      0x0400    /* bitA : 空      き          */
//      equ      0x0800    /* bitB : 空      き          */
//      equ      0x1000    /* bitC : 空      き          */
//      equ      0x2000    /* bitD : 空      き          */
//      equ      0x4000    /* bitE : 空      き          */
//      equ      0x8000    /* bitF : 空      き          */

/*****
/*
/*      SGDM immidiate Definition
/*
/*
*****/
/*-----*/
/*      CSW ( control select switch )
/*-----*/
#define ICLR      0x0001    /* bit.0 : Integration clear bit          */
#define ISEL      0x0002    /* bit.1 : Voltage compensation select bit */
//      --      0x0004    /* bit.2 : ----
//      --      0x0008    /* bit.3 : ----
#define OBSSEL     0x0008    /* bit.3 : Current Observer Select bit ; <V038> <V076> */
#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 Pass filter2 Disable       */
#define OVMSEL1    0x0100    /* bit.8 : Over modulation select(bit 8,9) */
#define OVMSEL2    0x0200    /* bit.9 : if bit8,9 = 10 or 11 then OVMOD2 */
//      //      if bit8,9 = 01 then OVMOD1
#define OVMMOD     0x0400    /* bit.A : modulation modification enable */
#define V_FB       0x0800    /* bit.B : 電 圧F B 方 式 弱 め界磁選択 */
#define DIDTSEL    0x1000    /* bit.C : L di/dt cmp select(1:di/dt,0:def) */
//      ANTIWU      0x2000    /* bit.D : anti-windup test
#define V_FB2      0x2000    /* bit.D : 電 圧F B 方 式 弱 め界磁選択2 <V531>
#define INT_ST     0x4000    /* bit.E : 飽 和 時 積 分停止
#define F3DSABL    0x8000    /* bit.F : Notch filter3 Disable
/*-----*/

```

---

---

```

/*  CPORT read data      bit address 4a0h      */
/*-----*/
//      --      0x0001  /* bit.0 :      */
//      --      0x0002  /* bit.1 :      */
//      --      0x0004  /* bit.2 :      */
#define RLOCK  0x0008  /* bit.3 : Transer register lock status      */
#define BB      0x0010  /* bit.4 : Base block bit      */
/*-----*/
/*      status set      bit      */
/*-----*/
//      --      0x0004  /* bit.2 :      */
#define DLIM  0x0008  /* bit.3 : D-axis integration limit status      */
#define QLIM  0x0008  /* bit.3 : Q-axis integration limit status      */
//      --      0x0010  /* bit.4 :      */
#define TLIM  0x0020  /* bit.5 : Torque Limit status      */
#define IHOLD 0x0040  /* bit.6 : Anti-windup test      */
//      --      0x0080  /* bit.7 :      */
/*-----*/
/*      status clear      bit      */
/*-----*/
//      --      0xffffb  /* bit.2 :      */
#define DLIMI 0xffff7  /* bit.3 :      */
#define QLIMI 0xffff7  /* bit.3 :      */
//      --      0xffef  /* bit.4 :      */
#define TLIMI 0xffdf  /* bit.5 :      */
#define IHOLDI 0xffbf  /* bit.6 : Anti-windup test      */
//      NON      0xff7f  /* bit.7 :      */
/*-----*/
/*      static value      */
/*-----*/
#define PI2      0x4000  /* 90deg      */
#define PI23     0x5555  /* 120deg     */

```

---

---

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