

```

/*****
/*
/*
/*  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FB演算処理 */

/*****
/*
/*      H/W Register                                */
/*
/*****
/*-----*/
/*      READ register                                */
/*-----*/
/*リードレジスタ追加(暫定)、定義名は別途対応<6.06>ここから */
#define MRD0 *(VUSHORT*)0x700 /* M-portスレーブ受信データワード0(0,1バイト) */
#define MRD4 *(VUSHORT*)0x704 /* M-portスレーブ受信データワード4(8,9バイト) */
#define MRD5 *(VUSHORT*)0x705 /* M-portスレーブ受信データワード5(10,11バイト) */
/*-----*/
#define MRD47 *(VUSHORT*)0x72F /* M-portスレーブ受信データワード47(94,95バイト) */
/*-----*/
#define SFT1R0 *(VUSHORT*)0x780 /* RC命令シフト1 保持レジスタ0(bit0-4) */
#define SFT1R1 *(VUSHORT*)0x781 /* RC命令シフト1 保持レジスタ1(bit0-4) */
#define SFT1R2 *(VUSHORT*)0x782 /* RC命令シフト1 保持レジスタ2(bit0-4) */
#define SFT1R3 *(VUSHORT*)0x783 /* RC命令シフト1 保持レジスタ3(bit0-4) */

```

---

```

#define SFT1R4 *(VUSHORT*)0x784 /* RC命令 シ フ ト 1 保持レジスタ4(bit0-4) */
#define SFT1R5 *(VUSHORT*)0x785 /* RC命令 シ フ ト 1 保持レジスタ5(bit0-4) */
#define SFT1R6 *(VUSHORT*)0x786 /* RC命令 シ フ ト 1 保持レジスタ6(bit0-4) */
#define SFT1R7 *(VUSHORT*)0x787 /* RC命令 シ フ ト 1 保持レジスタ7(bit 0 -4) R ound専用 */
#define SFT1R8 *(VUSHORT*)0x788 /* RC命令 シ フ ト 1 保持レジスタ8(bit 0 -4) R ound専用 */
#define SFT1R9 *(VUSHORT*)0x789 /* RC命令 シ フ ト 1 保持レジスタ9(bit0-4) */
#define SFT1R10 *(VUSHORT*)0x78A /* RC命令 シ フ ト 1 保持レジスタ10(bit0-4 ) */
#define SFT1R11 *(VUSHORT*)0x78B /* RC命令 シ フ ト 1 保持レジスタ11(bit0-4 ) */
#define SFT1R12 *(VUSHORT*)0x78C /* RC命令 シ フ ト 1 保持レジスタ12(bit0-4 ) */
#define SFT1R13 *(VUSHORT*)0x78D /* RC命令 シ フ ト 1 保持レジスタ13(bit0-4 ) */
#define SFT1R14 *(VUSHORT*)0x78E /* RC命令 シ フ ト 1 保持レジスタ14(bit0-4 ) */
#define SFT1R15 *(VUSHORT*)0x78F /* RC命令 シ フ ト 1 保持レジスタ15(bit0-4 ) */
/*-----*/
#define SFT2R0 *(VUSHORT*)0x790 /* RC命令 シ フ ト 2 保持レジスタ0(bit0-5) */
#define SFT2R1 *(VUSHORT*)0x791 /* RC命令 シ フ ト 2 保持レジスタ1(bit0-5) */
#define SFT2R2 *(VUSHORT*)0x792 /* RC命令 シ フ ト 2 保持レジスタ2(bit0-5) */
#define SFT2R3 *(VUSHORT*)0x793 /* RC命令 シ フ ト 2 保持レジスタ3(bit0-5) */
#define SFT2R4 *(VUSHORT*)0x794 /* RC命令 シ フ ト 2 保持レジスタ4(bit0-5) */
#define SFT2R5 *(VUSHORT*)0x795 /* RC命令 シ フ ト 2 保持レジスタ5(bit0-5) */
#define SFT2R6 *(VUSHORT*)0x796 /* RC命令 シ フ ト 2 保持レジスタ6(bit0-5) */
#define SFT2R7 *(VUSHORT*)0x797 /* RC命令 シ フ ト 2 保持レジスタ7(bit 0 -5) R ound専用 */
#define SFT2R8 *(VUSHORT*)0x798 /* RC命令 シ フ ト 2 保持レジスタ8(bit 0 -5) R ound専用 */
#define SFT2R9 *(VUSHORT*)0x799 /* RC命令 シ フ ト 2 保持レジスタ9(bit0-5) */
#define SFT2R10 *(VUSHORT*)0x79A /* RC命令 シ フ ト 2 保持レジスタ10(bit0-5 ) */
#define SFT2R11 *(VUSHORT*)0x79B /* RC命令 シ フ ト 2 保持レジスタ11(bit0-5 ) */
#define SFT2R12 *(VUSHORT*)0x79C /* RC命令 シ フ ト 2 保持レジスタ12(bit0-5 ) */
#define SFT2R13 *(VUSHORT*)0x79D /* RC命令 シ フ ト 2 保持レジスタ13(bit0-5 ) */
#define SFT2R14 *(VUSHORT*)0x79E /* RC命令 シ フ ト 2 保持レジスタ14(bit0-5 ) */
#define SFT2R15 *(VUSHORT*)0x79F /* RC命令 シ フ ト 2 保持レジスタ15(bit0-5 ) */
/*-----*/
#define NCTCTM *(VUSHORT*)0x7AC /* 指令用NCTタイマカウンタ値 */
#define NCTCFB *(VUSHORT*)0x7AD /* 指令用NCTフイー ド バックカウンタ値 */
#define NCTCCD *(VUSHORT*)0x7AE /* 指令用NCTCDウンタ値 */
#define NCTCTP *(VUSHORT*)0x7AF /* 指令用NCTTPウンタ値 */
/*-----*/
#define IREG_S *(VUSHORT*)0x7B0 /* 間接転送ソ ー スアドレス保持レジスタ設定 */
#define IREG_D *(VUSHORT*)0x7B1 /* 間接転送宛先アドレス保持レジスタ設定 */

```

```

#define DBSTS *(VUSHORT*)0x7B4 /* デッドバンドカウンタオーバー/アンダフロー */
#define DBU1 *(VUSHORT*)0x7B5 /* PU1Vデッドバンド補償カウンタラッチ値 */
#define DBV1 *(VUSHORT*)0x7B6 /* PV1Vデッドバンド補償カウンタラッチ値 */
#define DBW1 *(VUSHORT*)0x7B7 /* PW1Vデッドバンド補償カウンタラッチ値 */
#define ACMC4 *(VUSHORT*)0x7B8 /* 指令マールチプレクスチャンネル4 */
#define ACMC5 *(VUSHORT*)0x7B9 /* 指令マールチプレクスチャンネル5 */
#define ACMC6 *(VUSHORT*)0x7BA /* 指令マールチプレクスチャンネル6 */
#define ACMC7 *(VUSHORT*)0x7BB /* 指令マールチプレクスチャンネル7 */
#define FCCFZRNL *(VUSHORT*)0x7BE /* フルクローイズドカウンタ FZRNラッチデータ */
#define FCLDL *(VUSHORT*)0x7BF /* フルクローイズドカウンタロードラッチデータ */
/*-----*/
// #define UPDO *(VUSHORT*)0x7C0 /* 拡張 I/O 出力データ */
#define UPDI *(VUSHORT*)0x7C1 /* 拡張 I/O 入力データ */
#define SADAT_A *(VUSHORT*)0x7C2 /* エンコーダ AD データ A */
#define SADAT_AA *(VUSHORT*)0x7C3 /* エンコーダ AD データ AA */
#define SADAT_B *(VUSHORT*)0x7C4 /* エンコーダ AD データ B */
#define SADAT_BB *(VUSHORT*)0x7C5 /* エンコーダ AD データ BB */
#define FBZRND *(VUSHORT*)0x7C8 /* フォワードバックカウンタ ZRN ラッチデータ */
#define FBCLT *(VUSHORT*)0x7C9 /* フォワードバックカウンタロードラッチデータ */
#define FBPUd *(VUSHORT*)0x7CA /* フォワードバックカウンタ PU ラッチデータ */
#define FBPCD *(VUSHORT*)0x7CB /* フォワードバックカウンタ PC ラッチデータ */
#define TCNT *(VUSHORT*)0x7CC /* フォワードバックタイマ現在値 */
#define T1LT *(VUSHORT*)0x7CD /* フォワードバックタイマ A/B パルスラッチ値 */
#define T2LT *(VUSHORT*)0x7CE /* フォワードバックキャリアラッチ値 */
#define T3LT *(VUSHORT*)0x7CF /* フォワードバックタイマ T1 直前値ラッチ値 */
/* リードレジスタ追加 (暫定)、定義名は別途対応<6.06> ここまで */
/*-----*/
#define IuAD *(VUSHORT*)0x07d0 /* ADR:0x07d0: AD data U */
#define IvAD *(VUSHORT*)0x07d1 /* ADR:0x07d1: AD data V */
// --; /* ADR:0x07d2: */
/*-----*/
#define IFLAG *(VUSHORT*)0x07d3 /* ADR:0x07d3: Flag status */
// bit.0 : Z(zero)
// bit.1 : S(sign, '-')
// bit.2 : C(carry)
// bit.3 : B(borrow)
// 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)
/*-----*/
#define MACCL ((DWREG*)(VUSHORT*)0x07d4)->1 /* ADR:0x07d4: MPLY 0-31 bit */
#define MACCLL ((DWREG*)(VUSHORT*)0x07d4)->s[0] /* ADR:0x07d4: MPLY 0-15 bit */
#define MACCLH ((DWREG*)(VUSHORT*)0x07d4)->s[1] /* ADR:0x07d4: MPLY 16-31 bit */
/*-----*/
#define MACCH ((DWREG*)(VUSHORT*)0x07d6)->1 /* ADR:0x07d6: MPLY 32-63 bit */
#define MACCHL ((DWREG*)(VUSHORT*)0x07d6)->s[0] /* ADR:0x07d6: MPLY 32-47 bit */
#define MACCHH ((DWREG*)(VUSHORT*)0x07d6)->s[1] /* ADR:0x07d6: MPLY 48-63 bit */
/*-----*/
#define DivRem *(VUSHORT*)0x07d8 /* ADR:0x07d8: Divide modulo */
#define HSUR0 *(VUSHORT*)0x07d9 /* ADR:0x07d9: ホ ス ト 指 令 ポー ト 0 */
#define HSUR1 *(VUSHORT*)0x07da /* ADR:0x07da: ホ ス ト 指 令 ポー ト 1 */
#define HSUR2 *(VUSHORT*)0x07db /* ADR:0x07db: ホ ス ト 指 令 ポー ト 2 */
#define HSUR3 *(VUSHORT*)0x07dc /* ADR:0x07dc: ホ ス ト 指 令 ポー ト 3 */
/*-----*/
#define CTSTR *(VUSHORT*)0x07dd /* 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
/*-----*/
#define INT1SET *(VUSHORT*)0x07de /* 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
/*-----*/
#define FLTSTAT *(VUSHORT*)0x07df /* 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 : OVDAT(OV)
        // bit.5 : OCDAT(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)
/*-----*/
#define FBCSET *(VUSHORT*)0x07e0 /* 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 : PCDAT(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

```

---

---

```

/*-----*/
#define FCCST *(VUSHORT*)0x07e1 /* 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)

/*-----*/
#define CRUDP *(VUSHORT*)0x07e2 /* 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: */
#define OPCCLM *(VUSHORT*)0x07e4 /* ADR:0x07e4: OP counter CLM latch */
// --; /* ADR:0x07e5: */
// --; /* ADR:0x07e6: */
#define OPCLT *(VUSHORT*)0x07e7 /* ADR:0x07e7: OP counter load latch */
// --; /* ADR:0x07e8: */
// --; /* ADR:0x07e9: */
#define SRPG1RD2 *(VUSHORT*)0x07ea /* ADR:0x07ea: Serial-PG1 receive word2 */
#define SRPG1RD3 *(VUSHORT*)0x07eb /* ADR:0x07eb: Serial-PG1 receive word3 */
#define SRPG1RD4 *(VUSHORT*)0x07ec /* ADR:0x07ec: Serial-PG1 receive word4 */
#define SRPG1RD5 *(VUSHORT*)0x07ed /* ADR:0x07ed: Serial-PG1 receive word5 */
#define SRPG1RD6 *(VUSHORT*)0x07ee /* ADR:0x07ee: Serial-PG1 receive word6 */
#define SRPG1RD7 *(VUSHORT*)0x07ef /* ADR:0x07ef: Serial-PG1 receive word7 */

/*-----*/
#define SFTMACCL ((DWREG*) (VUSHORT*)0x07f0)->1 /* ADR:0x07f0: BRL shifter 0-31 bit */
#define SFTMACCLL ((DWREG*) (VUSHORT*)0x07f0)->s[0] /* ADR:0x07f0: BRL shifter 0-15 bit */
#define SFTMACCLH ((DWREG*) (VUSHORT*)0x07f0)->s[1] /* ADR:0x07f0: BRL shifter 16-31 bit */

/*-----*/
#define SFTMACCH ((DWREG*) (VUSHORT*)0x07f2)->1 /* ADR:0x07f2: BRL shifter 0-31 bit */

```

---

---

```

#define SFTMACCHL ((DWREG*)(VUSHORT*)0x07f2)->s[0] /* ADR:0x07f2: BRL shifter 0-15 bit */
#define SFTMACCHH ((DWREG*)(VUSHORT*)0x07f2)->s[1] /* ADR:0x07f2: BRL shifter 16-31 bit */
/*-----*/
#define CAD0D *(VUSHORT*)0x07f4 /* ADR:0x00f4: Operation AD0 data */
#define CAD1D *(VUSHORT*)0x07f5 /* ADR:0x00f5: Operation AD1 data Low */
#define CAD1H *(VUSHORT*)0x07f6 /* ADR:0x00f6: Operation AD1 data high */
// --; /* ADR:0x00f7: */
/*-----*/
#define ILR *(VUSHORT*)0x07f8 /* ADR:0x07f8: interrupt level read */
// bit.0-2 : interrupt0
// bit.4-6 : interrupt1
// bit.8-A : interrupt2
// bit.C-E : interrupt3
/*-----*/
// --; /* ADR:0x07f9: */
#define SRPGORD2 *(VUSHORT*)0x07fa /* ADR:0x07fa: serial PG0 Rx word 2 */
#define SRPGORD3 *(VUSHORT*)0x07fb /* ADR:0x07fb: serial PG0 Rx word 3 */
#define SRPGORD4 *(VUSHORT*)0x07fc /* ADR:0x07fc: serial PG0 Rx word 4 */
#define SRPGORD5 *(VUSHORT*)0x07fd /* ADR:0x07fd: serial PG0 Rx word 5 */
#define SRPGORD6 *(VUSHORT*)0x07fe /* ADR:0x07fe: serial PG0 Rx word 6 */
#define SRPGORD7 *(VUSHORT*)0x07ff /* ADR:0x07ff: serial PG0 Rx word 7 */
//;
/*-----*/
/* WRITE register */
/*-----*/
#define UPDO *(VUSHORT*)0x07c0 /* ADR:0x07c0: 拡張IO for debug <V720> */
/*-----*/
#define OUTPT *(VUSHORT*)0x07d0 /* ADR:0x07d0: INT2 port */
#define WDT1L *(VUSHORT*)0x07d1 /* ADR:0x07d1: WDT trigger port(WDT1L) */
/*-----*/
#define BBSET *(VUSHORT*)0x07d2 /* 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)
/*-----*/
#define CRST *(VUSHORT*)0x07d3 /* 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)
#define SOD0L *(VUSHORT*)0x07d4 /* ADR:0x07d4: Sync-Serial0 Low */
#define SOD0H *(VUSHORT*)0x07d5 /* ADR:0x07d5: Sync-Serial0 High */
#define SOD1L *(VUSHORT*)0x07d6 /* ADR:0x07d6: Sync-Serial1 Low */
#define SOD1H *(VUSHORT*)0x07d7 /* ADR:0x07d7: Sync-Serial1 High */
/*-----*/
#define SDMECLR *(VUSHORT*)0x07d8 /* 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
/*-----*/
#define ADSYNC *(VUSHORT*)0x07d9 /* ADR:0x07d9: AD sync */
// --; /* ADR:0x07da: */
/*-----*/
#define PWMOS *(VUSHORT*)0x07db /* ADR:0x07db: PWM out select */
        // 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)
/*-----*/
#define CrSet1 *(VUSHORT*)0x07dc /* ADR:0x07dc: Carrer setting */
        // bit.0-2 : T&B counter clear set
/*-----*/
#define CTSTW *(VUSHORT*)0x07dd /* 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

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

/*-----*/
#define PCVS0 *(VUSHORT*)0x07fa /* ADR:0x07fa: PWM-pulse convert set0 */
#define PCVS1 *(VUSHORT*)0x07fb /* ADR:0x07fb: PWM-pulse convert set1 */
#define PCVS2 *(VUSHORT*)0x07fc /* ADR:0x07fc: PWM-pulse convert set2 */
#define EIX *(VUSHORT*)0x07fd /* ADR:0x07fd: interrupt enable */
#define DIX *(VUSHORT*)0x07fe /* 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 0000h を強制セット要求 */
#define POSETCMDFF 0x0002 /* パルス変換位置に FFFFh 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 \*\*\*\*\*/