```
/*****************************************************************************************************/
/*                                                   */
/*                                                   */
/*    MprgSgdvStruct.h : S   G   D   V   マ   イ   ク   ロ   プ   ログラム構造 体定義                  */
/*                                                   */
/*                                                   */
/*****************************************************************************************************/
/*                                                   */
/*                                                   */
/*                                                   */
/*                                                   */
/*                                                   */
/*                                                   */
/*                                                   */
/*                                                   */
/*                                                   */
/*                                                   */
/************** Copyright (C) Yaskawa Electric Corporation ***********************************/
/*                                                   */
/*    Rev.1.00 : 2011.09.19  Y.Tanaka    ・   マ   イ   ク   ロ   用   構   造   体 定義 ファイル 作成          */
/*                                                   */
/*****************************************************************************************************/

/*****************************************************************************************************/
/*                                                   */
/*    Work Register : WREGBASE address = 08007000h                        */
/*                                                   */
/*****************************************************************************************************/
/*---------------------------------------------------------------------------------------------------*/
/*    32 bit registers                                        */
/*---------------------------------------------------------------------------------------------------*/
typedef struct{
  DWREG IdIntgl;          /* ADR:0x0002: (WREGBASE+004) ; ACRd Integral(Low)          */
  DWREG IqIntgl;          /* ADR:0x0004: (WREGBASE+008) ; ACRq Integral(Low)          */
  DWREG VdFil;          /* ADR:0x0006: (WREGBASE+00C) ; ACRd Filter Output(Low)       */
  DWREG VqFil;          /* ADR:0x0008: (WREGBASE+010) ; ACRq Filter Output(Low)       */
}ACRV;
/*---------------------------------------------------------------------------------------------------*/
```

```
/*    16 bit registers                                          */
/*--------------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT BbSetW;          /* ADR:0x000c: (WREGBASE+018) ; soft BB set(BBSET)          */
  SHORT FltStsW;         /* ADR:0x000d: (WREGBASE+01A) ; fault status(FLTIN)         */
  SHORT CtrlStsRW;         /* ADR:0x000e: (WREGBASE+01C) ; controll flag/status(CTSTR,CTSTW) */
  SHORT CNTSTS;          /* ADR:0x0016: (WREGBASE+02C) ; counter status(FBCSTS)        */
  SHORT FccStsMon;         /* ADR:0x0017: (WREGBASE+02E) ; full closed counter status(FCCST) */
  SHORT IntglFlg;          /* ADR:0x0080: (WREGBASE+100) ; 積　分　飽和ﾌﾗｸﾞ             */
}STSFLG;
/*--------------------------------------------------------------------------------------------------*/
/*    counter                                                 */
/*--------------------------------------------------------------------------------------------------*/
//<0.02>delete 未　使　用　のため
//typedef struct{
//  SHORT NCnt;          /* ADR:0x0018: (WREGBASE+030) ; FB count              */
//  SHORT DNCnt;           /* ADR:0x0019: (WREGBASE+032) ; dNCNT               */
//}CNTV;
/*--------------------------------------------------------------------------------------------------*/
/*    voltage FF                                              */
/*--------------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT LdC;           /* ADR:0x001a: (WREGBASE+034) ; Ld  * omega             */
  SHORT LqC;           /* ADR:0x001b: (WREGBASE+036) ; Lq  * omega             */
  SHORT MagC;          /* ADR:0x001c: (WREGBASE+038) ; Phi * omega             */
/*--------------------------------------------------------------------------------------------------*/
  SHORT VdOut;             /* ADR:0x0027: (WREGBASE+04E) ; Vd output voltage         */
  SHORT VqOut;             /* ADR:0x0028: (WREGBASE+050) ; Vq output voltage         */
  SHORT VdComp;          /* ADR:0x0029: (WREGBASE+052) ; Vd compensation data        */
  SHORT VqComp;          /* ADR:0x002a: (WREGBASE+054) ; Vq compensation data        */
  SHORT VuOut;             /* ADR:0x002b: (WREGBASE+056) ; vu output data (to PWM)       */
  SHORT VvOut;             /* ADR:0x002c: (WREGBASE+058) ; vv output data (to PWM)       */
  SHORT VwOut;             /* ADR:0x002d: (WREGBASE+05A) ; vw output data (to PWM)       */
/*--------------------------------------------------------------------------------------------------*/
  SHORT VdRef;           /* ADR:0x0034: (WREGBASE+068) ; vdref               */
  SHORT VqRef;           /* ADR:0x0035: (WREGBASE+06A) ; vqref               */
/*--------------------------------------------------------------------------------------------------*/
```

```
  SHORT Vmax2;          /* ADR:0x0074: (WREGBASE+0E8) ; 電　圧　指　令　ベ　ク　トル最大値(8192* 1.27) <V531>  */
  SHORT V12;          /* ADR:0x0075: (WREGBASE+0EA) ; 電　圧　指　令　ベ クトル(√(VdOut^2+VqOut^ 2))  <V531>  */
}VCMPV;
/*-----------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT SinT;         /* ADR:0x001d: (WREGBASE+03A) ; sin theta             */
  SHORT CosT;         /* ADR:0x001e: (WREGBASE+03C) ; cos theta             */
  SHORT SinT2;          /* ADR:0x001f: (WREGBASE+03E) ; sin(theta + 2pi/3)          */
  SHORT CosT2;          /* ADR:0x0020: (WREGBASE+040) ; cos(theta + 2pi/3)          */
  SHORT SinT3;          /* ADR:0x0021: (WREGBASE+042) ; sin(theta - 2pi/3)          */
  SHORT CosT3;          /* ADR:0x0022: (WREGBASE+044) ; cos(theta - 2pi/3)          */
}SINTBL;
/*-----------------------------------------------------------------------------------------------*/
/*    for notch filter output                                    */
/*-----------------------------------------------------------------------------------------------*/
//    IQO1L;          /* ADR:0x0036: (WREGBASE+06C) ; Filter1 output(Low)     ; <V388> 削　除   */
//    IQO1;         /* ADR:0x0037: (WREGBASE+06E) ; Filter1 output(High)      ; <V388> 削　除   */
//    IQO2L;          /* ADR:0x0038: (WREGBASE+070) ; Filter2 output(Low)     ; <V388> 削　除   */
//    IQO2;         /* ADR:0x0039: (WREGBASE+072) ; Filter2 output(High)      ; <V388> 削　除   */
/*-----------------------------------------------------------------------------------------------*/
/*    for notch filter1/2 work(input previous)                          */
/*-----------------------------------------------------------------------------------------------*/
//    IQIP;         /* ADR:0x003a: (WREGBASE+074) ; previous IQIN       ; <V388> 削　除   */
//    IQIPP;          /* ADR:0x003b: (WREGBASE+076) ; previous IQIP       ; <V388> 削　除   */
//    IQI2P;          /* ADR:0x003c: (WREGBASE+078) ; previous IQO1       ; <V388> 削　除   */
//    IQI2PP;         /* ADR:0x003d: (WREGBASE+07A) ; previous IQI2P       ; <V388> 削　除   */
/*-----------------------------------------------------------------------------------------------*/
/*    for notch filter1 work(output previous)                           */
/*-----------------------------------------------------------------------------------------------*/
//    IQO1P;          /* ADR:0x003e: (WREGBASE+07C) ; Filter1 output previous(Low)  ; <V388> 削　除   */
//    IQO1PH;         /* ADR:0x003f: (WREGBASE+07E) ; Filter1 output previous(High); <V388> 削　除  */
/*-----------------------------------------------------------------------------------------------*/
/*    for notch filter2 work(output previous)                           */
/*-----------------------------------------------------------------------------------------------*/
//DWREG IQOP;         /* ADR:0x0048: (WREGBASE+090) ; Filter2 output previous(Low)    *///110913tanaka21削　除（未　使　用のため）

/*-----------------------------------------------------------------------------------------------*/
```

```
/*    A/D Stop Error detect                                    */
/*-------------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT ADRst;          /* ADR:0x004c: (WREGBASE+098) ;                  */
/*-------------------------------------------------------------------------------------------------*/
/*    A/D Stop Error detect                                    */
/*-------------------------------------------------------------------------------------------------*/
  SHORT ADERRCNT;       /* ADR:0x004d: (WREGBASE+09A) ; ASIC AD Error Counter       */
  SHORT IRIUP;          /* ADR:0x004e: (WREGBASE+09C) ; Previous IRIU          */
  SHORT IRIUCNT;        /* ADR:0x004f: (WREGBASE+09E) ; same IRIU counter        */
  SHORT IRIVP;          /* ADR:0x0050: (WREGBASE+0A0) ; Previous IRIV          */
  SHORT IRIVCNT;        /* ADR:0x0051: (WREGBASE+0A2) ; same IRIV counter        */
}ADSTOP;
/*-------------------------------------------------------------------------------------------------*/
/*    reserved                                          */
/*-------------------------------------------------------------------------------------------------*/
SHORT WREG82;         /* ADR:0x0052: Work  Register 82                 */
SHORT WREG83;         /* ADR:0x0053: Work  Register 83                 */
SHORT WREG84;         /* ADR:0x0054: Work  Register 84                 */
SHORT WREG85;         /* ADR:0x0055: Work  Register 85                 */
SHORT WREG86;         /* ADR:0x0056: Work  Register 86                 */
SHORT WREG87;         /* ADR:0x0057: Work  Register 87                 */
SHORT WREG88;         /* ADR:0x0058: Work  Register 88                 */
SHORT WREG89;         /* ADR:0x0059: Work  Register 89                 */
/*-------------------------------------------------------------------------------------------------*/
/*    deferential                                       */
/*-------------------------------------------------------------------------------------------------*/
//VDDFL,VQDFLは IdDataP,IqDataP と 同 じ レ ジ ス タ で あ り 初 期 化 以外に参照していないため不要
//DWREG WREG90;         /* ADR:0x005a: Work  Register 90                   */
//#define VDDFL   WREG90.1  /*                                   */
//#define VDDFH   WREG90.s[1] /*                                    */
//DWREG WREG92;         /* ADR:0x005c: Work  Register 92                   */
//#define VQDFL   WREG92.1  /*                                   */
//#define VQDFH   WREG92.s[1] /*                                    */
SHORT WREG95;         /* ADR:0x005f: Work  Register 95                 */
SHORT WREG100;        /* ADR:0x0064: (WREGBASE+0C8);Work  Register 100         */
SHORT WREG101;        /* ADR:0x0065: Work  Register 101                */
```

```
SHORT WREG102;        /* ADR:0x0066: Work  Register 102                */
SHORT WREG103;        /* ADR:0x0067: Work  Register 103                */
SHORT WREG104;        /* ADR:0x0068: Work  Register 104                */
SHORT WREG109;        /* ADR:0x006d: Work  Register 109 ←   使  っ  て ない        */
/*-------------------------------------------------------------------------------------------*/
/*    for INT_HOST                                          */
/*-------------------------------------------------------------------------------------------*/
DWREG HTMP0;          /* ADR:0x0082: (WREGBASE+104) ; HOST Interrupt Temp0(130)     */
DWREG HTMP2;          /* ADR:0x0084: HOST Interrupt Temp2(132)              */
DWREG HTMP4;          /* ADR:0x0086: HOST Interrupt Temp4(134)              */
SHORT HTMP6;          /* ADR:0x0088: HOST Interrupt Temp6(136)              */
SHORT HTMP7;          /* ADR:0x0089: HOST Interrupt Temp7(137)              */
DWREG HTMP8;          /* ADR:0x008a: HOST Interrupt Temp8(138)              */
/*-------------------------------------------------------------------------------------------*/
/*    for ROUND                                         */
/*-------------------------------------------------------------------------------------------*/
SHORT WREG140;        /* ADR:0x008c: (WREGBASE+118) ; Work  Register 140       */
SHORT WREG141;        /* ADR:0x008d: Work  Register 141                */
SHORT WREG142;        /* ADR:0x008e: Work  Register 142                */
SHORT WREG143;        /* ADR:0x008f: Work  Register 143                */
SHORT WREG144;        /* ADR:0x0090: Work  Register 144                */
SHORT WREG145;        /* ADR:0x0091: Work  Register 145                */
SHORT WREG146;        /* ADR:0x0092: Work  Register 146                */
SHORT WREG147;        /* ADR:0x0093: Work  Register 147                */
SHORT WREG148;        /* ADR:0x0094: Work  Register 148                */
SHORT WREG149;        /* ADR:0x0095: Work  Register 149                */
/*-------------------------------------------------------------------------------------------*/
/*    for INT_AD                                          */
/*-------------------------------------------------------------------------------------------*/
DWREG TMP0;         /* ADR:0x0096: (WREGBASE+12C)                  */
DWREG TMP2;         /* ADR:0x0098: (WREGBASE+130)                  */
DWREG TMP4;         /* ADR:0x009a: (WREGBASE+134)                  */
DWREG TMP6;         /* ADR:0x009c: (WREGBASE+138)                  */
DWREG TMP8;         /* ADR:0x009e: (WREGBASE+13C)                  */
/*-------------------------------------------------------------------------------------------*/
/*    CPUか  ら  の  受  け  取 りレジス タ                          */
/*-------------------------------------------------------------------------------------------*/
```

```
typedef struct{
  SHORT IuOffsetIn;       /* ADR:0x00a0: (WREGBASE+140) ; A/D transfer offset for iu      */
  SHORT IvOffsetIn;       /* ADR:0x00a1: (WREGBASE+142) ; A/D transfer offset for iv      */
  SHORT KcuIn;          /* ADR:0x00a2: (WREGBASE+144) ; A/D transfer gain for iu      */
  SHORT KcvIn;          /* ADR:0x00a3: (WREGBASE+146) ; A/D transfer gain for iv      */
  SHORT IdIn;         /* ADR:0x00a4: (WREGBASE+148) ; Id reference            */
  SHORT IqIn;         /* ADR:0x00a5: (WREGBASE+14A) ; Iq reference            */
  SHORT PhaseHIn;       /* ADR:0x00a6: (WREGBASE+14C) ; PHA input             */
  SHORT VelIn;          /* ADR:0x00a7: (WREGBASE+14E) ; VEL input             */
/*-------------------------------------------------------------------------------------------*/
/*    voltage compensation                                   */
/*-------------------------------------------------------------------------------------------*/
  SHORT VdRefIn;          /* ADR:0x00bc: (WREGBASE+178) ; vdref input             */
  SHORT VqRefIn;          /* ADR:0x00bd: (WREGBASE+17A) ; vqref input             */
/*-------------------------------------------------------------------------------------------*/
/*    Torque Limit                                       */
/*-------------------------------------------------------------------------------------------*/
  SHORT TLimPIn;        /* ADR:0x00c8: (WREGBASE+190) ; +tlim               */
  SHORT TLimMIn;        /* ADR:0x00c9: (WREGBASE+192) ; -tlim               */
}ADINV;
/*-------------------------------------------------------------------------------------------*/
/*    static variable                                    */
/*-------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT Kcu;          /* ADR:0x0011: (WREGBASE+022) ; A/D transfer gain for iu      */
  SHORT Kcv;          /* ADR:0x0012: (WREGBASE+024) ; A/D transfer gain for iv      */
/*-------------------------------------------------------------------------------------------*/
/*    for over modulation                                    */
/*-------------------------------------------------------------------------------------------*/
  SHORT Kmod;           /* ADR:0x004a: (WREGBASE+094) ; Over modulation gain/offset     */
  SHORT Kvv;          /* ADR:0x004b: (WREGBASE+096) ; AVR                 */
/*-------------------------------------------------------------------------------------------*/
  SHORT Kc;           /* ADR:0x00a8: (WREGBASE+150) ; Current conversion Gain       */
  SHORT Ld;           /* ADR:0x00a9: (WREGBASE+152) ; d axis Inductance         */
  SHORT Lq;           /* ADR:0x00aa: (WREGBASE+154) ; q axis Inductance         */
  SHORT Mag;          /* ADR:0x00ab: (WREGBASE+156) ; Magnetic flux (Phi)         */
  SHORT KdP;          /* ADR:0x00ac: (WREGBASE+158) ; d axis propotion gain (PI control)  */
```

```
  SHORT KqP;          /* ADR:0x00ad: (WREGBASE+15A) ; q axis propotion gain (PI control)  */
  SHORT KdI;          /* ADR:0x00ae: (WREGBASE+15C) ; d axis integral time (gain)     */
  SHORT KqI;          /* ADR:0x00af: (WREGBASE+15E) ; q axis integral time (gain)     */
  SHORT VdLim;          /* ADR:0x00b0: (WREGBASE+160) ; d axis integral limit       */
  SHORT VqLim;          /* ADR:0x00b1: (WREGBASE+162) ; q axis integral limit       */
  SHORT KvvIn;          /* ADR:0x00b2: (WREGBASE+164) ; Voltage Compasation gain      */
  SHORT OnDelayLvl;         /* ADR:0x00b3: (WREGBASE+166) ; On delay change level(97.5.26 mo) */
  SHORT Tfil;         /* ADR:0x00b4: (WREGBASE+168) ; Filter time constant        */
  SHORT Vmax;         /* ADR:0x00b5: (WREGBASE+16A) ; voltage limit data (Vmax^2)     */
  SHORT OnDelayComp;      /* ADR:0x00b6: (WREGBASE+16C) ; On-delay compensation       */
  SHORT CtrlSw;         /* ADR:0x00b7: (WREGBASE+16E) ; Control Flag           */
  SHORT CrFreq;         /* ADR:0x00b8: (WREGBASE+170) ; carrier freq reference(change while BB) */
//    --;           /* ADR:0x00b9: (WREGBASE+172) ;                  */
//    --;           /* ADR:0x00ba: (WREGBASE+174) ;                  */
//    --;           /* ADR:0x00bb: (WREGBASE+176) ;                  */
/*-------------------------------------------------------------------------------------------------*/
/*    notch1                                       */
/*-------------------------------------------------------------------------------------------------*/
  SHORT Kf11;         /* ADR:0x00be: (WREGBASE+17C)                */
  SHORT Kf12;         /* ADR:0x00bf: (WREGBASE+17E)                */
  SHORT Kf13;         /* ADR:0x00c0: (WREGBASE+180)                */
  SHORT Kf14;         /* ADR:0x00c1: (WREGBASE+182)                */
/*-------------------------------------------------------------------------------------------------*/
/*    notch2                                       */
/*-------------------------------------------------------------------------------------------------*/
  SHORT Kf21;         /* ADR:0x00c2: (WREGBASE+184)                */
  SHORT Kf22;         /* ADR:0x00c3: (WREGBASE+186)                */
  SHORT Kf23;         /* ADR:0x00c4: (WREGBASE+188)                */
  SHORT Kf24;         /* ADR:0x00c5: (WREGBASE+18A)                */
/*-------------------------------------------------------------------------------------------------*/
/*    notch3                                       */
/*-------------------------------------------------------------------------------------------------*/
  SHORT Kf31;         /* ADR:0x0069: (WREGBASE+0D2)                */
  SHORT Kf32;         /* ADR:0x006a: (WREGBASE+0D4)                */
  SHORT Kf33;         /* ADR:0x006b: (WREGBASE+0D6)                */
  SHORT Kf34;         /* ADR:0x006c: (WREGBASE+0D8)                */
/*-------------------------------------------------------------------------------------------------*/
```

```
/*    LPF                                                */
/*-------------------------------------------------------------------------------------------------*/
  SHORT TLpf;          /* ADR:0x00c6: (WREGBASE+18C) ; LPF1 gain              */
  SHORT TLpf2;           /* ADR:0x00c7: (WREGBASE+18E) ; LPF2 gain              */
/*-------------------------------------------------------------------------------------------------*/
  SHORT MotResist;         /* ADR:0x00ca: (WREGBASE+194) ; Moter resistance(r1/I_BASE*V_BASE)  */
  SHORT OnDelaySlope;      /* ADR:0x00cb: (WREGBASE+196) ; OnDelay Comp. slope gain      */
//    --;              /* ADR:0x00cc: (WREGBASE+198) ;                    */
//    --;              /* ADR:0x00cd: (WREGBASE+19A) ;                    */
  SHORT L_dIdt;          /* ADR:0x00ce: (WREGBASE+19C) ;                      */
/*-------------------------------------------------------------------------------------------------*/
  SHORT FccRst;          /* ADR:0x00cf: (WREGBASE+19E) ; FCCST Reset(for TEST)       */
}INTADP;


typedef struct{
  SHORT IuOffset;        /* ADR:0x000f: (WREGBASE+01E) ; A/D transfer offset for iu      */
  SHORT IvOffset;        /* ADR:0x0010: (WREGBASE+020) ; A/D transfer offset for iv      */
/*-------------------------------------------------------------------------------------------------*/
  SHORT IuInData;          /* ADR:0x0023: (WREGBASE+046) ; iu input data (from A/D)      */
  SHORT IvInData;          /* ADR:0x0024: (WREGBASE+048) ; iv input data (from A/D)      */
  SHORT IdInData;          /* ADR:0x0025: (WREGBASE+04A) ; Id Input              */
  SHORT IqInData;          /* ADR:0x0026: (WREGBASE+04C) ; Iq Input              */
/*-------------------------------------------------------------------------------------------------*/
//    --;           /* ADR:0x002e: (WREGBASE+05C) ;                    */
/*-------------------------------------------------------------------------------------------------*/
  SHORT CrFreqW;           /* ADR:0x002f: (WREGBASE+05E) ; Carrier freq Now          */
/*-------------------------------------------------------------------------------------------------*/
  SHORT IuOut;             /* ADR:0x0030: (WREGBASE+060) ; Iu reference(for dead-time cmp)   */
  SHORT IvOut;             /* ADR:0x0031: (WREGBASE+062) ; Iv reference(for dead-time cmp)   */
/*-------------------------------------------------------------------------------------------------*/
  SHORT V1;            /* ADR:0x0032: (WREGBASE+064) ; V1( = SQRT(VdOut^2+VqOut^2) )       */
  SHORT Vcent;         /* ADR:0x0033: (WREGBASE+066) ; Central voltage           */
/*-------------------------------------------------------------------------------------------------*/
/*    for LPF                                          */
/*-------------------------------------------------------------------------------------------------*/
  DWREG IqOut1Lpf;           /* ADR:0x0040: (WREGBASE+080) ; LPF1 Output(Low)          */
```

```
  DWREG IqOut2Lpf;          /* ADR:0x0042: (WREGBASE+084) ; LPF2 Output(Low)          */
/*-------------------------------------------------------------------------------------------*/
  SHORT IqRef;          /* ADR:0x0044: (WREGBASE+088) ; Iq Reference after limit(5/23)    */
  SHORT TLimP;          /* ADR:0x0045: (WREGBASE+08A) ; +tlim 5/23              */
  SHORT TLimM;          /* ADR:0x0046: (WREGBASE+08C) ; -tlim 5/23              */
/*-------------------------------------------------------------------------------------------*/
/*    for monitor                                        */
/*-------------------------------------------------------------------------------------------*/
  SHORT IqMon;          /* ADR:0x0047: (WREGBASE+08E) ; IQ monitor              */
/*-------------------------------------------------------------------------------------------*/
  SHORT IdDataP;        /*                                    */
  SHORT IqDataP;        /*                                    */
  SHORT KEangle;        /* ADR:0x005e: (0BC)  ; 機  械  角  →  電  気 角変換係 数<V013 >        */
  DWREG IdLfil;         /* ADR:0x0060: L(di/dt)フ  ィ  ルタ                    */
  DWREG IqLfil;         /* ADR:0x0062:                              */
/*-------------------------------------------------------------------------------------------*/
//--- ノ  ッ  チ  フ  ィ ルタ出力 ---
//    IqOut3L;          /* ADR:0x006e: (WREGBASE+0DC) ; notch filter output(Low)      ; <V388> 削  除   */
//    IqOut3;         /* ADR:0x006f: (WREGBASE+0DE) ; notch filter output(High)     ; <V388> 削  除   */
//--- ノ  ッ  チ  フ  ィ  ル タ用ワーク ---
//    IQI3P;          /* ADR:0x0070: (WREGBASE+0E0) ; previous IQO1           ; <V388> 削  除   */
//    IQI3PP;         /* ADR:0x0071: (WREGBASE+0E2)   ; previous IQI3P          */
//    IQO3P;          /* ADR:0x0072: (WREGBASE+0E4) ; notch filter output previous(Low) ; <V388> 削  除 */
//    IQO3PH;         /* ADR:0x0073: (WREGBASE+0E6) ; notch filter output previous(High); <V388> 削  除 */
/*-------------------------------------------------------------------------------------------*/
//    --;           /* ADR:0x0076: (WREGBASE+0EC)                 */
//    --;           /* ADR:0x0077: (WREGBASE+0EE)                 */
//    --;           /* ADR:0x0078: (WREGBASE+0F0)                 */
//    --;           /* ADR:0x0079: (WREGBASE+0F2)                 */
//    --;           /* ADR:0x007a: (WREGBASE+0F4)                 */
//    --;           /* ADR:0x007b: (WREGBASE+0F6)                 */
//    --;           /* ADR:0x007c: (WREGBASE+0F8)                 */
//    --;           /* ADR:0x007d: (WREGBASE+0FA)                 */
//    --;           /* ADR:0x007e: (WREGBASE+0FC)                 */
//    --;           /* ADR:0x007f: (WREGBASE+0FE)                 */
//    --;           /* ADR:0x0081: (WREGBASE+102)                 */
/*-------------------------------------------------------------------------------------------*/
```

```
/*    for axis q monitor        <V224>                              */
/*--------------------------------------------------------------------------------------------------*/
  SHORT IqDistIn;       /* ADR:0x0200: q軸  外  乱  ト  ルク入力              ADDR = 08007400 */
  SHORT IqDist;         /* ADR:0x0201: q軸  外  乱  トルク                  ADDR = 08007402 */
  SHORT IqMonFil;       /* ADR:0x0202: q軸  電  流  指  令 モ  ニ  タ(フィル タ 後 )        ADDR = 08007404 */
  SHORT IqOfRef;        /* ADR:0x0203: q軸  電  流 指  令  ( 外  乱 トルク加 算 後 )        ADDR = 08007406 */
/*--------------------------------------------------------------------------------------------------*/
/*    for notch filter1         <V388>                              */
/*--------------------------------------------------------------------------------------------------*/
  DWREG IqOut1L;          /* ADR:0x0250: filter1 output(Low)            ADDR = 080074A0 */
  DWREG IqOut1PL;         /* ADR:0x0252: filter1 output 1delay(Low)       ADDR = 080074A4 */
  DWREG IqOut1PPL;        /* ADR:0x0254: filter1 output 2delay(Low)       ADDR = 080074A8 */
  DWREG IqIn1PL;          /* ADR:0x0256: filter1 input 1delay(Low)          ADDR = 080074AC */
  DWREG IqIn1PPL;         /* ADR:0x0258: filter1 input 2delay(Low)          ADDR = 080074B0 */
  //    IQO1;         /* ADR:0x026a: filter1 output(short type)       ADDR = 080074B4   // <V502> */
  SHORT IQO1;         /* ADR:0x025a: filter1 output(short type)       ADDR = 080074B4   // <V502> */
  DWREG IqOut1BufL;       /* ADR:0x025c: filter1 output buffer(Low)       ADDR = 080074B8   // <V502> */
/*--------------------------------------------------------------------------------------------------*/
/*    for notch filter2         <V388>                              */
/*--------------------------------------------------------------------------------------------------*/
  DWREG IqOut2L;          /* ADR:0x0260: filter2 output(Low)            ADDR = 080074C0 */
  DWREG IqOut2PL;         /* ADR:0x0262: filter2 output 1delay(Low)       ADDR = 080074C4 */
  DWREG IqOut2PPL;        /* ADR:0x0264: filter2 output 2delay(Low)       ADDR = 080074C8 */
  DWREG IqIn2PL;          /* ADR:0x0266: filter2 input 1delay(Low)          ADDR = 080074CC */
  DWREG IqIn2PPL;       /* ADR:0x0268: filter2 input 2delay(Low)          ADDR = 080074D0 */
  SHORT IQO2;         /* ADR:0x026a: filter2 output(short type)       ADDR = 080074D4 */
  DWREG IqOut2BufL;       /* ADR:0x026c: filter2 output buffer(Low)       ADDR = 080074D8   // <V502> */
/*--------------------------------------------------------------------------------------------------*/
/*    for notch filter3         <V388>                              */
/*--------------------------------------------------------------------------------------------------*/
  DWREG IqOut3L;          /* ADR:0x0270: filter3 output(Low)            ADDR = 080074E0 */
  DWREG IqOut3PL;         /* ADR:0x0272: filter3 output 1delay(Low)       ADDR = 080074E4 */
  DWREG IqOut3PPL;        /* ADR:0x0274: filter3 output 2delay(Low)       ADDR = 080074E8 */
  DWREG IqIn3PL;          /* ADR:0x0276: filter3 input 1delay(Low)          ADDR = 080074EC */
  DWREG IqIn3PPL;       /* ADR:0x0278: filter3 input 2delay(Low)          ADDR = 080074F0 */
  SHORT IqOut3;         /* ADR:0x027a: filter3 output(short type)       ADDR = 080074F4 */
  DWREG IqOut3BufL;       /* ADR:0x027c: filter3 output buffer(Low)       ADDR = 080074F8   // <V502> */
```

```
}INTADV;

typedef struct{
  USHORT  Low;              /* 引  数  下位16 bit値                    */
  USHORT  High;             /* 引  数  上位16 bit値                    */
  USHORT  uswk0;              /* 平  方  根  演算用 １６ｂｉｔワ ークレジスタ0          */
  USHORT  uswk1;              /* 平  方  根  演算用 １６ｂｉｔワ ークレジスタ1          */
  USHORT  uswk3;              /* 平  方  根  演算用 １６ｂｉｔワ ークレジスタ3          */
  USHORT  uswk4;              /* 平  方  根  演算用 １６ｂｉｔワ ークレジスタ4          */
  USHORT  uswk5;              /* 平  方  根  演算用 １６ｂｉｔワ ークレジスタ5          */
  USHORT  uswk6;              /* 平  方  根  演算用 １６ｂｉｔワ ークレジスタ6          */
  ULONG ulwk0;              /* 平  方  根  演算用 ３２ｂｉｔワ ークレジスタ0          */
  ULONG ulwk2;              /* 平  方  根  演算用 ３２ｂｉｔワ ークレジスタ2          */
  DWREG tmp0;             /* 平  方  根  演算用16/ ３２ｂｉｔワ ークレジスタ0       */
}SQRTWK;

typedef struct{
  SHORT swk0;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ0            */
  SHORT swk1;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ1            */
  SHORT swk2;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ2            */
  SHORT swk3;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ3            */
  SHORT swk4;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ4            */
  SHORT swk5;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ5            */
  SHORT swk6;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ6            */
  SHORT swk7;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ7            */
  SHORT swk8;             /* 電  流  制 御用１６ｂｉｔワ  ークレジスタ8            */
  CSHORT* pCtbl;              /* 電  流  制  御  用  テ  ー  ブ  ル  ポ  イ ンタ用ワークレ ジ ス タ      */
  LONG  lwk0;             /* 電  流  制 御用３２ｂｉｔワ  ークレジスタ0            */
  LONG  lwk1;             /* 電  流  制 御用３２ｂｉｔワ  ークレジスタ1            */
  LONG  lwk2;             /* 電  流  制 御用３２ｂｉｔワ  ークレジスタ2            */
  LONG  lwk4;             /* 電  流  制 御用３２ｂｉｔワ  ークレジスタ4            */
  LONG  lwk6;             /* 電  流  制 御用３２ｂｉｔワ  ークレジスタ6            */
  LONG  lwk8;             /* 電  流  制 御用３２ｂｉｔワ  ークレジスタ8            */
  SQRTWK  sqrtwk;             /* 平  方  根  演  算  用  ワーク構造体              */
}INTADWK;

/*-------------------------------------------------------------------------------------------*/
```

```
/*    for Current Observer                                    */
/*------------------------------------------------------------------------------------------------*/
// パ ラ メ ータ用
typedef struct{
  SHORT TsPerL;          /* ADR:0x00d0: Ts/L ( オ ブ ザ ー バ 用 パラ メ ー タ )      ADDR = 080071A0 */
  SHORT Gobs;          /* ADR:0x00d1: g ( オ ブ ザ ー バの極 )          ADDR = 080071A2 */
  SHORT RLTs;          /* ADR:0x00d2: 1-R･Ts/L ( オ ブ ザ ー バ 用 パ ラ メータ )   ADDR = 080071A4 */
  SHORT FilObsGain;          /* ADR:0x00d3: フ ィ ル タ ゲイン              ADDR = 080071A6 */
}DOBSP;
// 計 算 用
typedef struct{
  DWREG LpfIlq;          /* ADR:0x00d4: q軸 ロ ー パ ス フ ィルタ変数 ( L ow byte )   ADDR = 080071A8 */
  DWREG HpfIlq;          /* ADR:0x00d6: q軸 ハ イ パ ス フ ィルタ変数 ( L ow byte )   ADDR = 080071AC */
  SHORT IqObsOut;          /* ADR:0x00d8: q軸 オ ブ ザ ーバ出力              ADDR = 080071B0 */
  SHORT IqObsFreq;          /* ADR:0x00d9: q軸 振 動 成分                ADDR = 080071B2 */
  DWREG LpfIld;          /* ADR:0x00da: d軸 ロ ー パ ス フ ィルタ変数 ( L ow byte )   ADDR = 080071B4 */
  DWREG HpfIld;          /* ADR:0x00dc: d軸 ハ イ パ ス フ ィルタ変数 ( L ow byte )   ADDR = 080071B8 */
  SHORT IdObsOut;          /* ADR:0x00de: d軸 オ ブ ザ ーバ出力              ADDR = 080071BC */
  SHORT IdObsFreq;          /* ADR:0x00df: d軸 振 動 成分                ADDR = 080071BE */
  SHORT DmpGain;          /* ADR:0x00e0: 電 流 オ ブ ザ ー バ ダ ンピング ゲ イ ン     ADDR = 080071C0 */
}DOBSV;


/*------------------------------------------------------------------------------------------------*/
/*    for Phase Interpolate     <V112>                          */
/*------------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT PHAL;          /* ADR:0x000a: (WREGBASE+014) ; Phase(Low)              */
  SHORT PhaseH;          /* ADR:0x000b: (WREGBASE+016) ; Phase(High)              */
  SHORT PhaseIp;          /* ADR:0x0100: 位 相 補 間量                ADDR = 08007200 */
  SHORT PhaseIpF;          /* ADR:0x0101: 位 相 補 間 フ ラ グ (0 : しない 、 1 :する)   ADDR = 08007202 */
  SHORT PhaseIpIn;          /* ADR:0x0102: 位 相 補 間量 (CPU→ A S I C受け 渡 し 用)     ADDR = 08007204 */
  SHORT PhaseIpFIn;          /* ADR:0x0103: 位 相 補 間 フラ グ(CP U → A SI C受 け 渡し用)   ADDR = 08007206 */
}PHASEV;

/*------------------------------------------------------------------------------------------------*/
/*    for Encoder IF                                <V720>  */
```

```
/*-------------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT AmpType;         /* ADR:0x0110: Rotary Type or Linear Type       ADDR = 08007220 */
/*-------------------------------------------------------------------------------------------------*/
  DWREG RcvPosX0;        /* ADR:0x0112: Motor Encoder Position(受  信  位置) Low     ADDR = 08007224 */
  DWREG RcvPosX1;        /* ADR:0x0114: Motor Encoder Position(受  信  位置) Low     ADDR = 08007228 */
  DWREG RcvPosX2;        /* ADR:0x0116: Motor Encoder Position(受  信  位置) Low     ADDR = 0800722C */
/*-------------------------------------------------------------------------------------------------*/
  SHORT MotPosSftR;        /* ADR:0x0118: Shift Data for Motor Position Cal.    ADDR = 08007230 */
  SHORT MotPosSftX;        /* ADR:0x0119: Shift Data for Motor Position Cal.    ADDR = 08007232 */
  SHORT MotPosSign;        /* ADR:0x011a: Sign  Data for Motor Position Cal.    ADDR = 08007234 */
  SHORT DivOutSft;         /* ADR:0x011b: 分  周  出  力  シフト数              ADDR = 08007236 */
  DWREG AccErrLv;        /* ADR:0x011c: Motor Encoder Acc. Error Check Low   ADDR = 08007238 */
/*-------------------------------------------------------------------------------------------------*/
  DWREG DivPls;          /* ADR:0x011e: 分  周  出  力 パルス Low             ADDR = 0800723C */
/*-------------------------------------------------------------------------------------------------*/
  DWREG DivOutGain;        /* ADR:0x0120: 分  周  出  力 ゲ  イ ン(リニア)  Lo w       ADDR = 08007240 */
  DWREG DivPos;          /* ADR:0x0122: 分  周  出  力 パ  ル ス(リニア)  Lo w       ADDR = 08007244 */
  DWREG DivPlsRem;          /* ADR:0x0124: 分  周  出  力  パ ル  ス 余り(リニ ア )  Low      ADDR = 08007248 */
/*-------------------------------------------------------------------------------------------------*/
  SHORT SPGFail;         /* ADR:0x0126: S-PG受  信  失  敗  許容回数          ADDR = 0800724C */
/*-------------------------------------------------------------------------------------------------*/
  SHORT BitIprm;         /* ADR:0x0127: Bit Initial Parameter           ADDR = 0800724E */
  SHORT BitData;           /* ADR:0x0128: Bit Data                  ADDR = 08007250 */
/*-------------------------------------------------------------------------------------------------*/
  SHORT IncPlsReq;         /* ADR:0x0129: PAO IncrePulse Output Request        ADDR = 08007252 */
  SHORT PAOSeqCmd;         /* ADR:0x012a: PAO Serial Output Sequence       ADDR = 08007254 */
  SHORT PlsOSetCmd;        /* ADR:0x012b: パ  ル  ス  出  力  回  路初期化要求          ADDR = 08007256 */
}ENCIFV;

/*-------------------------------------------------------------------------------------------------*/
/*    分  周  パ  ル  ス  出力関連：HostCPU  - - >  A si c 定周期転送用                 <V720>  */
/*-------------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT IncPlsReqIn;       /* ADR:0x0130: PAO IncrePulse Output Request        ADDR = 08007260 */
  SHORT PAOSeqCmdIn;       /* ADR:0x0131: PAO Serial Output Sequence       ADDR = 08007262 */
  SHORT PlsOSetCmdIn;        /* ADR:0x0132: パ  ル  ス  出  力  回  路初期化要求          ADDR = 08007264 */
```

```
/*-----------------------------------------------------------------------------------------------*/
  SHORT DivSetIn;       /* ADR:0x0133: 分 周 機 能 設定入力            ADDR = 08007266 */
  SHORT PoSet1In;       /* ADR:0x0134: パ ル ス 変 換 原点補正1          ADDR = 08007268 */
  SHORT PoSet2In;       /* ADR:0x0135: パ ル ス 変 換 原点補正2          ADDR = 0800726A */
/*-----------------------------------------------------------------------------------------------*/
  DWREG IncInitPlsIn;       /* ADR:0x0136: 初 期 イ ン ク レパルス Low          ADDR = 0800726C */
  DWREG IncInitRemIn;       /* ADR:0x0138: 初 期 イ ン ク レ パルス余り Lo w        ADDR = 08007270 */
/*-----------------------------------------------------------------------------------------------*/
  SHORT AccCntClrReq;     /* ADR:0x013a: 加 速 度 チ ェ ッ ク 開 始 カウン ト ク リア要求    ADDR = 08007274 */
/*-----------------------------------------------------------------------------------------------*/
/*    Register for Library                                <V720>  */
/*-----------------------------------------------------------------------------------------------*/
  DWREG Argu0;          /* ADR:0x01c0: Argument0                 ADDR = 08007380 */
  DWREG Argu1;          /* ADR:0x01c2: Argument1                 ADDR = 08007384 */
  SHORT Argu2;          /* ADR:0x01c4: Argument2                 ADDR = 08007388 */
  SHORT Argu2H;         /* ADR:0x01c5: Argument2 High Word           ADDR = 0800738A */
  DWREG Ret0;         /* ADR:0x01c6: Return Register             ADDR = 0800738C */
  DWREG Kx;           /* ADR:0x01c8: kx                  ADDR = 08007390 */
  DWREG Sx;           /* ADR:0x01ca: sx                  ADDR = 08007394 */
  DWREG Iu0;          /* ADR:0x01cc: iu[0]                   ADDR = 08007398 */
  SHORT Iu1;          /* ADR:0x01ce: iu[1]                   ADDR = 0800739C */
  SHORT Iu1H;         /* ADR:0x01cf: iu[1]                   ADDR = 0800739E */
}DIVPLSV;


/*-----------------------------------------------------------------------------------------------*/
/*    初 期 化 処 理 用ワ ーク                                  <V720>  */
/*-----------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT IN_WK0;         /* ADR:0x0140: 初 期 化 処理用 Work0           ADDR = 08007280 */
  SHORT IN_WK0H;        /* ADR:0x0141: Work0             ADDR = 08007282     */
  SHORT IN_WK1;         /* ADR:0x0142: Work1             ADDR = 08007284     */
  SHORT IN_WK1H;        /* ADR:0x0143: Work1             ADDR = 08007286     */
}INITWK;

/*-----------------------------------------------------------------------------------------------*/
/*    @INT_ENC用 ワ ー ク                                  <V720>  */
```

```
/*-------------------------------------------------------------------------------------------------*/
typedef struct{
  SHORT EncWk0;          /* ADR:0x0144: Encoder Interrup Work0          ADDR = 08007288 */
  SHORT EncWk0H;         /* ADR:0x0145: Work0          ADDR = 0800728A       */
  SHORT EncWk1;          /* ADR:0x0146: Work1          ADDR = 0800728C       */
  SHORT EncWk1H;         /* ADR:0x0147: Work1          ADDR = 0800728E       */
/*-------------------------------------------------------------------------------------------------*/
  DWREG RxPos;           /* ADR:0x0148: Encoder Position Data Low         ADDR = 08007290 */
  SHORT RxFlg0;          /* ADR:0x014a: エ ン コ ー ダ 受 信 チェックフ ラ グ          ADDR = 08007294 */
}IENCWK;

/*-------------------------------------------------------------------------------------------------*/
/*    分 周 パ ル ス 更新処理(@I N T _ HOST) 用ワー ク                          <V720>  */
/*-------------------------------------------------------------------------------------------------*/
typedef struct{
  USHORT  Divuswk;        /* 追 加(16/32bi t 兼 用 で 使 っ て い た ワーク レ ジ スタの分離 )     */
/*-------------------------------------------------------------------------------------------------*/
  LONG  DivWk0;          /* ADR:0x0150: 分 周 パ ル ス 更新処理用 Work0       ADDR = 080072A0 */
  LONG  DivWk1;          /* ADR:0x0152: Work1        ADDR = 080072A4         */
/*-------------------------------------------------------------------------------------------------*/
  LONG  RxPos0;        /* ADR:0x0154: Encoder Position Data Low          ADDR = 080072A8 */
  LONG  RcvPosX;         /* ADR:0x0156: Motor Encoder Position Low        ADDR = 080072AC */
  LONG  LastRcvPosX;         /* ADR:0x0158: Last Motor Encoder Position Low       ADDR = 080072B0 */
/*-------------------------------------------------------------------------------------------------*/
  LONG  DMotPos;         /* ADR:0x015a: Delta Motor Position Low          ADDR = 080072B4 */
/*-------------------------------------------------------------------------------------------------*/
  SHORT EncMstErrCnt;        /* ADR:0x015c: Encoder Mst. Error Counter        ADDR = 080072B8 */
  SHORT AccChkCnt;         /* ADR:0x015d: ADDR = 080072BA                    */
  LONG  MotAcc;          /* ADR:0x015e: Motor Acc. (2Word)           ADDR = 080072BC */
/*-------------------------------------------------------------------------------------------------*/
  LONG  IncInitPls;          /* ADR:0x0160: 初 期 イ ン ク レ パルス出力 Lo w         ADDR = 080072C0 */
/*-------------------------------------------------------------------------------------------------*/
  SHORT DivSetW;         /* ADR:0x0162: 分 周 機 能設定                 ADDR = 080072C4 */
  SHORT PoSet1W;         /* ADR:0x0163: パ ル ス 変 換 原点補正1            ADRR = 080072C6 */
  SHORT PoSet2W;         /* ADR:0x0164: パ ル ス 変 換 原点補正2            ADDR = 080072C8 */
/*-------------------------------------------------------------------------------------------------*/
  USHORT  uswk;          /* ホ ス ト 割 込み用 １６ｂｉｔ ワークレジスタ0             */
```

```
  SHORT swk0;          /* ホ  ス  ト  割 込み用  1 6 b i t ワークレジスタ1            */
  SHORT swk1;          /* ホ  ス  ト  割 込み用  1 6 b i t ワークレジスタ1            */
  LONG  lwk1;          /* ホ  ス  ト  割 込み用  3 2 b i t ワークレジスタ0            */
  LONG  lwk2;          /* ホ  ス  ト  割 込み用  3 2 b i t ワークレジスタ1            */
  LONG  lwk3;          /* ホ  ス  ト  割 込み用  3 2 b i t ワークレジスタ2            */
}IHOSTWK;


/*-------------------------------------------------------------------------------------------------*/
/*    Register for Library                                  <V720>  */
/*-------------------------------------------------------------------------------------------------*/
typedef struct{
  LONG  lswk1;          /* ADR:0x01d2: swk1 for Library            ADDR = 080073A4 */
  LONG  lswk2;          /* ADR:0x01d4: swk2 for Library            ADDR = 080073A8 */
  LONG  lswk3;          /* ADR:0x01d6: swk3 for Library            ADDR = 080073AC */
  LONG  lswk4;          /* ADR:0x01d8: wk4 for Library             ADDR = 080073B0 */
  LONG  lswk10;         /* ADR:0x01e4: wk10 for Library            ADDR = 080073C8 */
  LONG  lswk11;         /* ADR:0x01e6: wk11 for Library            ADDR = 080073CC */
  DLREG dlwk;         /* 64bit計  算  用  ワ  ー  クレジスタ                    */
}IPFBWK;

/*-------------------------------------------------------------------------------------------------*/
/*    for Weaken Field    <V214>                                */
/*-------------------------------------------------------------------------------------------------*/
typedef struct{
//----- CPUか  ら  渡  さ  れる変数 -----
  SHORT WfKpVLIn;       /* ADR:0x0300: 電  圧F B 比  例 ゲ  イン(下位16 bi t)          ADDR = 08007600 */
  SHORT WfKpVHIn;       /* ADR:0x0301: 電  圧F B 比  例 ゲ  イン(上位16 bi t)          ADDR = 08007602 */
  SHORT WfKiVLIn;       /* ADR:0x0302: 電  圧F B 積  分 ゲ  イン(下位16 bi t)          ADDR = 08007604 */
  SHORT WfKiVHIn;       /* ADR:0x0303: 電  圧F B 積  分 ゲ  イン(上位16 bi t)          ADDR = 08007606 */
  SHORT WfV1MaxIn;        /* ADR:0x0304: 電  圧  指  令  制 限レベル              ADDR = 08007608 */
  SHORT WfIdRefLimIn;     /* ADR:0x0305: d軸  電  流  指  令 リミット             ADDR = 0800760A */
//
//----- CPUか  ら  初  期  化  時  に 渡される変数  -- -- -
  SHORT WfIntegLim;       /* ADR:0x0306: 電  圧F B 積  分  リミット           ADDR = 0800760C */
//
//----- マ  イ  ク  ロ  で 扱う変数 -----
```

```
  SHORT IdOut;          /* ADR:0x0013: (WREGBASE+026) ; Id reference            */
  SHORT IqOut;          /* ADR:0x0014: (WREGBASE+028) ; Iq reference            */
  SHORT Vel;          /* ADR:0x0015: (WREGBASE+02A) ; Velocity (omega)          */
//    欠  番;         /* ADR:0x0307:                                 */
  DWREG WfKpV;          /* ADR:0x0308: 電  圧F B 比  例 ゲ  イン(下位16 bi t)            ADDR = 08007610 */
  DWREG WfKiV;          /* ADR:0x030a: 電  圧F B 積  分 ゲ  イン(下位16 bi t)            ADDR = 08007614 */
  SHORT WfV1Max;        /* ADR:0x030c: 電  圧  指  令  制 限レベル              ADDR = 08007618 */
  SHORT WfIdRefLim;       /* ADR:0x030d: d軸  電  流  指  令 リミット             ADDR = 0800761A */
  SHORT WfVqMax;        /* ADR:0x030e: q軸  電  圧  指  令 リミット             ADDR = 0800761C */
//    欠  番;         /* ADR:0x030f:                                 */
  DWREG WfIntgl;        /* ADR:0x0310: 電  圧F B 積 分値                  ADDR = 08007620 */
  SHORT WfVdRef;        /* ADR:0x0312: d軸  電  圧 指令                 ADDR = 08007624 */
  SHORT WfVqRef;        /* ADR:0x0313: q軸  電  圧 指令                 ADDR = 08007626 */
}WEAKFV;

/*-----------------------------------------------------------------------------------------------*/
/*    マ  イ  ク  ロ  演  算  部  ソ  フ トバ  ー  ジ  ョ  ン  情 報  :   マイクロ演算 部→ホ ストCPU転送用        〈V720〉  */
/*-----------------------------------------------------------------------------------------------*/
SHORT MswVer;       /* ADR:0x0400: ソ  フ  ト  バ  ー  ジョン情報              ADDR = 08007800 */
SHORT TstVer;       /* ADR:0x0401: テ  ス  ト  バ  ー  ジョン情報              ADDR = 08007802 */
SHORT YspVer;       /* ADR:0x0402: Y  仕  様  バ  ー  ジョン情報              ADDR = 08007804 */
//#pragma AsicReg end


/***********************************************************************************************/
/*                                                 */
/*    定  数  定義                                      */
/*                                                 */
/***********************************************************************************************/
extern ACRV AcrV;                   /* 電  流  制  御 用変数                    */
extern STSFLG StsFlg;                 /* ス  テ  ー  タ  スフラグ                  */
extern VCMPV  VcmpV;                  /* 電  圧  補  償  演 算用変数                */
extern SINTBL SinTbl;                 /* sinテ  ー  ブル                    */
extern ADSTOP AdStop;                 /* 電  流  制  御  停  止  エ  ラ ー検出用構造体        */
extern ADINV  AdinV;                  /* ホ  ス トC P U 受  け 渡し変数             */
extern INTADP IntAdP;                 /* 電  流  割  込  み  パラメータ             */
extern INTADV IntAdV;                 /* 電  流  割  込 み変数                */
```

```
extern INTADWK  IntAdwk;                /* 電 流 割 込 みワーク                */
extern DOBSP  DobsP;                    /* D軸 オ ブ ザ ー バ 用パラメータ          */
extern DOBSV  DobsV;                    /* D軸 オ ブ ザ ー バ用変数              */
extern PHASEV PhaseV;                   /* 位 相 演 算 用変数                  */
extern ENCIFV EncIfV;                   /* エ ン コ ー ダ イ ン タフェース変数       */
extern DIVPLSV  DivPlsV;                /* 分 周 パ ル ス用変数                */
extern IPFBWK IPfbwk;                   /* 余 り 付 き位 置FB計算              */
extern IENCWK IEncWk;                   /* エ ン コ ー ダ 割 込みワーク           */
extern IHOSTWK  IHostWk;                /* ホ ス ト 割 込 みワーク              */
extern WEAKFV WeakFV;                   /* 弱 め 界 磁 用変数                  */
```