

```

/* $Header: P:/wip/software/pvcs/projects/d2000/inc/d1ktype.h_v
1.4 Sep 17
1999 12:04:28 lowell $ */
/*****
** FILE: D1KTYPE.H
**
** Copyright(c) 1997-present. All rights reserved.
** Biocontrol Technology Incorporated, Indiana, PA USA
** This file constitutes intellectual property of the company and
may
** not be used for other than company purposes.
**
** DESCRIPTION:
** External visible type definitions for D2000 data structures
.
**
** LIMITATIONS:
** None.
**
** SOFTWARE HISTORY:
** 07AUG97 5147 STR#0000 T. Kerr
** Initial creation.
** 11Jun99 5225 STR#0000 L.A.Wadsworth
** Corrected syntax error in blank single line comment
** 04Aug99 5225 STR#0000 L.A.Wadsworth
** Added MACHINE_INFO Structure
** 17Aug99 5225 STR#0000 L.A.Wadsworth
** Added MDM_READING Structure,
** added mode & count to DT_SPECTRUM structure and
** changed patient_id & instrument_id in DT_SPECTRUM to match s
peccs
** 18Aug99 5225 STR#0000 L.A.Wadsworth
** Cleaned up contents, added READING & END_OF_DATA to DT_Object
tID,
** and deleted DT_EVENT structure
** 31Aug99 5225 STR#0000 L.A.Wadsworth
** Changed serial_num & bico_User_id field size & type,
** deleted clinic_user_id field, added crc fields to selected s
tructures,
** added SPECTRUM_STATUS structure and SPECTRUM_STAT DT_ObjectI
D,
** changed instrument_id to serial_num everywhere,
** changed UINT to UWORD.
** 02Sep99 5225 STR#0000 L.A.Wadsworth
** Changed SECTOR_SIZE constants to long and
** changed DT_ObjectID END_OF_FILE back to END_OF_DATA.
** 14Sep99 5225 STR#0000 L.A.Wadsworth

```

```

** Added bico_logonid & bico_password to DT_ACCESS_INFO structu
re
** Adjusted DT_ACCESS_INFO to correct several word boundary
** alignment problems.
** Changed TELEMED crc field to crc_16.
** 15Sep99 5225 STR#0000 L.A.Wadsworth
** Changed TELEMED crc field to crc_16
** Changed TELEMED field array size values to named defined con
stants.
**
** 23SEP99 5147 SPR#0000 K.Pasumarthu
** Added a DT_CommandCode "CMD_GET_MCDATA" to retrieve data fro
m
** the modem card.
**
** 27SEP99 5147 SPR#0000 K.Pasumarthu
** Added a DT_CommandCode "PLOT" to plot the spectra collected
and
** added a DT_CommandCode "CLR_MDM_MEMORY" to clear the modem m
emory.
** Added two more fields(doubles) to the skin library
**
** 29SEP99 5225 SPR#0000 L.A.Wadsworth
** Added MD_MDM_DATE Structure for recording date in modem memo
ry
** that the sitting data is for.
** Added MD_PATIENT_STATUS Structure for saving the date of the
** last good sitting.
**
** 05OCT99 5225 SPR#0000 L.A.Wadsworth
** Corrected SKIN_LIB alterLimitLoEndpoint and alterLimitHiEndp
oint
** to alertLimitLoEndpoint and alertLimitHiEndp
oint
** 06OCT99 5225 SPR#0000 L.A.Wadsworth
** Corrected DT_CONTROL_COUNTERS gd_sittngs to gd_sittings.
**
** 08OCT99 5225 SPR#0000 S. F. NIZIOL
** Added structure DT_HEMO_AVG, define DT_HEMO_AVG_NUM and
** added and additional item to enum DT_BinCode
**
** 19OCT99 5147 SPR#0000 K.Pasumarthu
** Added two new object id's and 4 new commands for erasing 186
flash,
** obtaining and setting power on time and unit serial number.
**
** 20OCT99 5147 SPR#0000 K.Pasumarthu

```

```

**      Moved commands and associated structures to command.h
**
**      21OCT99  5147  SPR#0000 K.Pasumathy
**      Added a new structure DT_186_FLASH_SECTOR_0 which contains t
he
**      information stored in the first sector of the 186 flash.
**      22OCT99  5225  SPR#0000 S. F. NIZIOL
**      Changed spectrum and spectrum status structures--patient_id
**      is now bico_user_id to be consistent with user_info structur
e
**
**      28OCT99  5225  SPR#0000 V.Zhuze
**      Moved basic type definitions back from command.h,
**      moved DT_CommandFlag, DT_CommandStatus and DT_SETCLOCK_COMMA
ND
**      definitions to command.h
**
**      03NOV99  5225  SPR#0000 S. F. NIZIOL
**      Removed ctrlPredMatrix,DT_MAX_CTRL_PREDICT_RANKS,
**      numCtrlPredictRanks,controlScaling,meanCalControlPred and va
lidCtrlRange
**      and added ctrlRefMatrix and DT_CONTROL_ELEM for new control
check
**
**      03NOV99  5225  SPR#0000 L.A.Wadsworth
**      Added new Structure OPER_PARMS in DT_UNIT_INFO and DT_MACHIN
E_INFO
**      structures for machine operation control.
**      Added fill in DT_USER_ID after SKIN_LIBRARY to
**      align CONFIG_DATA to 28514.
**
**      05NOV99  5225  SPR#0000 L.A.Wadsworth
**      Added bicoProxyServer, bicoFtpServer, & spare_ipAddr.
**      Moved contents of PATIENT_STATUS structure into CONTROL_COUN
TERS.
**
**      16NOV99  5225  SPR#0000 S. F. NIZIOL
**      Added DT_ALERT structure, ALERT DT_ObjectID, changed DT_MDM_
DATE
**      to DT_EOD_DATE and changed MDM_DATE to EOD_DATE in the DT_Ob
jectID enum
**
**      18NOV99  5225  SPR#0000 L.A.Wadsworth
**      Revised PHONE_DATA structure to consolidate the phone codes
into two
**      fields: phoneNumber and alt_phoneNumber. The size of the PH
ONE_DATA
**      structure is unchanged.

```

```

** 19NOV99 5225 SPR#0000 L.A.Wadsworth
** changed DT_ACCESS_INFO ACfiller to
** isp_PAP_enable ISP Password Authentication service
**
** 22NOV99 5225 SPR#0000 V.Zhuze
** Changed enumeration types DT_ObjectID and Dt_Priority to DT_
UWORD type
** in the DT_OBJECT_HEADER and DT_QUEUE_ELEMENT type definitio
ns to make
** these structures scalable for 32-bit compiler platforms (e.
g., UNIX)
**
** 03DEC99 5225 SPR#0000 V.Zhuze
** Removed END_OF_DATA from DT_ObjectID enumeration type;
** commented out FAR macro definition for UNIX platform
**
** 08DEC99 5225 SPR#0000 S. F. NIZIOL
** Changed spectrum status field... error_Codes to err_code
**
** 10DEC99 5225 SPR#0000 V.Zhuze
** Changed the order of the fields in DT_SPECTRUM, DT_SPECTRUM_
STATUS,
** DT_READING_ID, and DT_ALERT to align 4-byte-long fields at
4 byte
** boundary (UNIX post-processing requirement); bico_user_id a
nd
** serial_num are now positioned right after the object header
**
** 14DEC99 5225 SPR#0000 V.Zhuze
** Added DT_UC_READINGS_ST constant to mark the start of readin
gs
** area on the user card
** 22DEC99 5225 SPR#0000 L.A.Wadsworth
** Added DT_TEL_USERCARD_ST & DT_TEL_TELEMED_ST for
** Telemedicine Download data structure location.
** 05Jan99 5225 SPR#0000 P. Roth
** Added misc_enables to DT_OPER_PARMS without changing size of
structure
** Added bit define DISP_HEMO_AVG for misc_enables
** 31JAN00 5225 SPR#0000 S. F. NIZIOL
** Added bit define DISP_PRED_AVG for misc_enables
**
** 03FEB00 5225 SPR#0000 L. A. Wadsworth
** Changed isp_AP_enable to isp_UU_enable for UU/ insertion to
isp_logonid.
**
** 08Feb00 5225 SPR#0000 P. Roth
** Renamed patient_hold to misc_enables which now is a bitfield

```

```

**    Removed misc_enables element from oper_parm structure.
**
**    23FEB00 5229 SPR#0000 S. F. NIZIOL
**    Added define for demo enable in misc_enables of DT_CONTROL_C
COUNTERS
*****
***/
/* ***
* Make sure the Visual C "Options/Project/Compiler/Preprocessor/
* Symbols and macros to define" has: _DEBUG, _DOS, EMBEDDED
*** */

#ifndef _D1KTYPE_H_
#define _D1KTYPE_H_

/****
* System include files
****/

/****
* Local include files
****/

/****
* Typedefs
****/

#if defined COMPILER_32_BIT
/****
* Definitions for 32-bit compilers
****/
typedef int          DT_DWORD;
typedef short        DT_INT;
typedef short        DT_INT16;
typedef int          DT_INT32;
typedef unsigned int  DT_TIME;
typedef unsigned int  DT_UDWORD;
typedef unsigned short DT_UINT16;
typedef unsigned int  DT_UINT32;
typedef unsigned short DT_UWORD;
typedef short        DT_WORD;

#else
/****
* Definitions for 16-bit compilers

```

```

****/
typedef long          DT_DWORD;    4 bytes 32
typedef int           DT_INT;      16 bits
typedef int           DT_INT16;    16 bits
typedef long          DT_INT32;    32 bits
typedef unsigned long DT_TIME;     0 32
typedef unsigned long DT_UDWORD;    0 32
typedef unsigned int  DT_UINT16;    0 16
typedef unsigned long DT_UINT32;    0 32
typedef unsigned int  DT_UWORD;     0 16
typedef int           DT_WORD;      16

```

DW

#endif

```

typedef char          DT_BOOL;      ✓
typedef char          DT_CHAR;      ✓
typedef double         DT_DOUBLE;   32
typedef double         DT_FLOAT;     32
typedef unsigned char DT_UCHAR;     8

```

```

/****
* Defines
****/

```

```

/****
* Object type identifiers.
****/

```

```

enum DT_ObjectID
{
    UNKNOWN = 0,
    COMMAND, 1
    DATA_PACKET, 2
    EVENT, 3
    SPECTRUM, 4
    READING, 5
    SPECTRUM_STATUS, 6
    EOD_DATE, 7
    POWER_ON_TIME, 8
    UNIT_SERIAL_NUMBER, 9
    ALERT, 10
    MACHINE_DATA_UPDT = 16,
    PATIENT_DATA_UPDT
};

```

4, 5, 6, 10

-32k + 32k

0 -

/****

```

* Defines
***/
#define DT_MAX_HIST_MEAS 2730*5
#define DT_MAX_SENSOR_CHANNELS 11
#define DT_MAX_SPECTRA_CHANNELS 64
#define DT_FALSE 0
#define DT_TRUE 1
#define DT_EPS 1.0e-20

/****
* Array size defines for machine, patient, & skin library
***/
#define DT_MAX_SIZ_BICO_USER_ID 10
#define DT_MAX_SIZ_UNIT_SERIAL_NUM 11
#define DT_MAX_SIZ_USER_NAME 17
#define DT_MAX_SKIN_PREDICT_RANKS 3
#define DT_MAX_OUTLIER_RANKS 45

/****
* Far pointer declaration for MS-DOS executables
***/
#if !defined(UNIX)
#if defined MSC_VER
#define FAR far
#elif defined __BORLANDC__
#define FAR __far
#else
#define FAR
#endif
#endif

/****
* Operating modes.
***/
enum DT_OperatingMode
{
    MODE_UNKNOWN = 0,
    MODE_CALIBRATION,
    MODE_EVALUATION,
    MODE_MEASUREMENT,
    MODE_RECALIBRATION,
    MODE_TEST
};

/****

```

* Bin codes.

enum DT_BinCode

{

 BIN_BEF_BREK = 1,

 BIN_AFT_BREK,

 BIN_BEF_LUN,

 BIN_AFT_LUN,

 BIN_BEF_DIN,

 BIN_AFT_DIN,

 BIN_BEDTIME,

 BIN_NIGHT,

 BIN_OTHER,

 BIN_CODE_LAST /* place after last bin code */

};

/**

* Spectrum types.

*/

enum DT_SpectrumType

{

 DT_UNKNOWN = 0,

 DT_CONTROL,

 DT_DARK,

 DT_INDEX, /* not really a spectrum type; used by protomgr for indexing */

 DT_REFERENCE,

 DT_SKIN,

 DT_CONTROL_ABSORBANCE,

 DT_DARK_ABSORBANCE,

 DT_REFERENCE_ABSORBANCE,

 DT_SKIN_ABSORBANCE,

 DT_SPECTRUM_STAT

};

/**

* Object priorities.

* Higher priorities have lower numbers.

*/

enum DT_Priority

{

 URGENT = 0,

 ROUTINE,

 LOW

};


```

/****
* Header to be placed as the first field in any object that can be
* queued.
****/
typedef struct
{
    DT_UWORD id; 4 - 2 bytes /* the object's unique identifier */
    DT_UWORD size; 2 bytes /* the object's size in bytes */
} DT_OBJECT_HEADER;

/****
* Calendar time structure
****/
typedef struct
{
    DT_UCHAR seconds; /* seconds, 0 - 59 */
    DT_UCHAR minutes; /* minutes, 0 - 59 */
    DT_UCHAR hours; /* hours, 0 - 23 */
    DT_UCHAR day_of_week; /* Sunday - Saturday, 1 - 7 */
    DT_UCHAR day_of_month; /* day of mon., 1 - 31 */
    DT_UCHAR month; /* month, 1 - 12 */
    DT_UWORD year; /* calendar year (1998, 1999, 2000, 2001, etc.) */
}
DT_RTC_TIME;

/****
* A queue element that links arbitrary object structures.
****/
typedef struct
{
    void* prev; /* pointer to previous queue element */
    void* next; /* pointer to next queue element */
    DT_UWORD priority; /* object priority; lower # = higher priority */
    DT_OBJECT_HEADER* objPtr; /* this pointer must be type cast to the specific object type(i.e., DT_ONEINT_COMMAND, DT_DATA_PACKET, DT_EVENT, etc.) */
}
DT_QUEUE_ELEMENT;

```

```

/**
 * A linked list of DT_QUEUE_ELEMENT structures.
 */
typedef struct
{
    DT_QUEUE_ELEMENT* first;    /* pointer to first element */
    DT_QUEUE_ELEMENT* last;    /* pointer to last element */
} DT_QUEUE;

/**
 * A raw spectral data object.
 */
typedef struct
{
    DT_OBJECT_HEADER objHdr;    /* the standard object header */
    DT_UWORD spectrum_type;    /* type of spectrum (ref, skin,
etc.) */
    DT_TIME time_stamp;        /* 10ms ticks since beginning o
f year */
    DT_UWORD year;            /* current year for time conver
sion */
    DT_UWORD dsp_config;        /* bit field for DSP and hardwa
re config */
    DT_UWORD alg_type;        /* Algorithm type */
    DT_UWORD integration_points; /* number of integration points
*/
    DT_UWORD num_channels;    /* number of channels to acquir
e data */
    union
    {
        DT_UINT32 pixels[DT_MAX_SPECTRA_CHANNELS]; /* spectral d
ata from ADC */
        DT_UCHAR bytes[DT_MAX_SPECTRA_CHANNELS * sizeof(DT_UINT3
2)];
    } spectrum;
    DT_UINT16 sensors[DT_MAX_SENSOR_CHANNELS]; /* sensor data f
rom ADC */
} DT_DATA_PACKET;

/**
 * A spectrum object in floating-point format.
 */
typedef struct
{

```

8/18/85

```

    DT_OBJECT_HEADER objHdr;          /* the standard object header */
/
    DT_UDWORD bico_user_id;          /* patient id corresponding to
this data */

    DT_UCHAR  serial_num[DT_MAX_SIZ_UNIT_SERIAL_NUM + 1];
/* instrument_id corresponding
to this data */
    DT_UDWORD spectrum_type;          /* type of spectrum (ref, skin,
etc.) */
    DT_UDWORD mode;                   /* Instrument operating mode */
    DT_UDWORD sit_count;              /* Spectrum sitting serial coun
t */
    DT_UDWORD ses_count;              /* Spectrum session serial coun
t */
    DT_UDWORD sub_count;              /* Spectrum subsession serial c
ount */
    DT_UDWORD year;                   /* current year for time conver
sion */
    DT_TIME   time_stamp;              /* 10ms ticks since beginning o
f year */

    DT_UDWORD bin_code;               /* corresponding bin code */
    DT_INT    invasive;               /* corresponding invasive meter
reading */
    DT_DOUBLE spectrum[DT_MAX_SPECTRA_CHANNELS];
    DT_DOUBLE sensors[DT_MAX_SENSOR_CHANNELS];
}
DT_SPECTRUM;

/****
* A spectrum status record.
****/
typedef struct
{
    DT_OBJECT_HEADER objHdr;          /* the standard object header */
/
    DT_UDWORD bico_user_id;          /* patient id corresponding to
this data */

    DT_UCHAR  serial_num[DT_MAX_SIZ_UNIT_SERIAL_NUM + 1];
/* instrument_id corresponding
to this data */
    DT_UDWORD spectrum_type;          /* type of spectrum (ref, skin,
etc.) */
    DT_UDWORD mode;                   /* Instrument operating mode */
    DT_UDWORD sit_count;              /* Spectrum sitting serial coun
t */

```

```

    DT_UWORD  ses_count;           /* Spectrum session serial coun
t */
    DT_UWORD  sub_count;           /* Spectrum subsession serial c
ount */
    DT_UWORD  year;                /* current year for time conver
sion */
    DT_TIME   time_stamp;          /* 10ms ticks since beginning o
f year */

    DT_UWORD  bin_code;            /* corresponding bin code */
    DT_INT    err_code;            /* error code for the sitting */
/
}
DT_SPECTRUM_STATUS;

/**
 * Skin library structures.
 ***/

typedef struct
{
    DT_DOUBLE sensorTol[DT_MAX_SENSOR_CHANNELS]; /* sensor chann
els, tolerance */
}
DT_CONFIG_DATA;

typedef struct
{
    DT_DOUBLE loEndpoint; /* low (minimum) for range */
    DT_DOUBLE hiEndpoint; /* high (maximum) for range */
}
DT_RANGE_ELEM;

typedef struct
{
    DT_UWORD  rankNum;
    DT_DOUBLE rankWeight;
    DT_DOUBLE calibConst;
    DT_DOUBLE calibVector[DT_MAX_SPECTRA_CHANNELS];
}
DT_RANK_ELEM;

typedef struct
{
    DT_RTC_TIME date;
    DT_DOUBLE  ctrlRefLimit;

```

```

    DT_DOUBLE    ctrlRefSensors[DT_MAX_SENSOR_CHANNELS - 1];
    DT_DOUBLE    ctrlRefVector[DT_MAX_SPECTRA_CHANNELS];
}
DT_CONTROL_ELEM;

typedef struct
{
    DT_UWORD      empty;           /* nonzero = structure empty, zero = nonempty */
    DT_RTC_TIME   time;           /* date and time information was stored */
    DT_INT        glucose;        /* measured glucose value */
    DT_INT        control;        /* measured control sample value */
    DT_INT        invasive;       /* invasive reading value */
    DT_INT        bin;            /* bin code for this reading */
    DT_INT        qm_flag;        /* quality monitoring indicator */
    DT_UWORD      status;         /* measurement algorithm completion status */
    DT_INT        correction;      /* bias correction value used to get measurement */
}
DT_READING;

typedef struct
{
    DT_UDWORD     bico_user_id;    /* BICO id for user, 4 integers + 6 integers */
    DT_UCHAR      serial_num[DT_MAX_SIZE_UNIT_SERIAL_NUM + 1];
                                   /* D2000 unit serial number */
    DT_UWORD      mode;           /* Instrument operating mode */
}
DT_READING_ID;

typedef struct
{
    DT_OBJECT_HEADER objHdr;      /* Standard object header */
    DT_READING_ID    readingID;   /* Machine, patient, & mode information */
    DT_READING       reading;     /* Evaluation or Measurement mode reading */
}

```

DT_MDM_READING;

typedef struct

```
{
    /* Delays are in number of 10 millisecond timer tics */
    DT_UDWORD ramEndedDelay;          /* Delay from end of ram movement to start of A/D reading */
    DT_UDWORD shutterClosedDelay;     /* Shutter closing time before dark reading, */
    /*
    DT_INT16 dark_IntegrationPoints; /* Dark integration points, currently 3640 */
    DT_INT16 ref_IntegrationPoints;   /* Reference integration points, currently 3640 */
    DT_INT16 ctrl_IntegrationPoints /* Control integration points, currently 3640 */
    /*;
    DT_INT16 skin_IntegrationPoints; /* Skin integration points, currently 910 */
    DT_UDWORD ditherEndedDelay;       /* Delay from end of dither movement to start of A/D reading, */
    DT_INT16 readsPerDither;          /* Number of skin readings per dither sequence, currently 16 */
    DT_INT16 firstRefReadings;        /* Number of first reference readings, currently one */
    DT_INT16 ctrlReadings;            /* Number of control readings, currently one */
    /*
    DT_INT16 secondRefReadings;       /* Number of second reference readings, currently one */
    DT_INT16 darkReadings;            /* Number of dark readings, currently one */
    DT_INT16 darkCorrectEnable;       /* Dark correction enable, currently enabled */
    /*
    DT_INT16 ditherSteps;              /* Dither steps, currently 120 */
    DT_INT16 ditherStepsHalfSpeed;    /* Half speed dither steps, currently 12 */
    DT_DOUBLE refRatioLoLimit;        /* Reference ratio check limits, currently 0.997 */
    */
}
```

```

    DT_DOUBLE refRatioHiLimit;      /* Reference ratio check li
mits, currently
1.003 */
    DT_INT16 skinLevelPixel;        /* Which pixel used for ski
n level check,
currently pixel 13 */
    DT_DOUBLE skinLevelPixelLoLimit; /* Pixel skin level check a
nd limit,
currently 0.3 */
    DT_DOUBLE skinLevelPixelHiLimit; /* Pixel skin level check a
nd limit,
currently 0.95 */
    DT_INT16 warmUpDelayEnabled;    /* Machine warm-up delay en
able, currently
not enabled */
    DT_UDWORD warmUpDelayTime;      /* Machine warm-up delay ti
me, default th 3
hours */
    DT_INT16 expansionFill1;        /* for future parameters */
    DT_INT16 expansionFill2;        /* for future parameters */
    DT_INT16 expansionFill3;        /* for future parameters */
    DT_INT16 expansionFill4;        /* for future parameters */
} DT_OPER_PARMS;

typedef struct
{
    DT_RTC_TIME build_time;          /* date and time info set a
t factory (build)
*/

    DT_UCHAR serial_num[DT_MAX_SIZ_UNIT_SERIAL_NUM + 1];
/* D2000 unit serial number
*/

    DT_UWORD mode;                  /* Instrument operating mod
e */

    DT_DOUBLE sensor_coeff[DT_MAX_SENSOR_CHANNELS];
/* environmental sensor
scaling coefficients */

    DT_UWORD sensor_algor[DT_MAX_SENSOR_CHANNELS];
/* Sensor scaling algorithm
*/

    DT_OPER_PARMS operParms;        /* machine operating parame
ters */

```

```

}
DT_UNIT_INFO;

typedef struct
{
    DT_UDWORD bico_user_id;      /* BICO id for user, 4 integers +
    6 integers */

    DT_UCHAR user_name[DT_MAX_SIZ_USER_NAME + 1]; /* display name
    for user,                                     ascii, NULL
    term */

    DT_RTC_TIME calib_time;      /* date and time of user calibrat
    ion */
}
DT_USER_INFO;

typedef struct
{
    DT_UNIT_INFO unit;           /* unit information */
    DT_USER_INFO curr_user;      /* current user information */
    DT_UDWORD   crc_16;          /* cyclic redundancy check for st
    ructure */
}
DT_USER_ID;

typedef struct
{
    DT_DWORD createDate;         /* creation date */

    DT_UDWORD numSpectraChan;     /* number of spectra channels
    */

    DT_UDWORD numOutlierRanks;    /* number of ranks in list fo
    r outlier                                detection (in loadings mat
    rix) */

    DT_UDWORD numSkinPredictRanks; /* number of ranks in list fo
    r skin                                prediction purpose(in Pre
    diction matrix)
    */

    DT_RANGE_ELEM rangeSensors[DT_MAX_SENSOR_CHANNELS];

```



```

ge */                                /* sensor channels, calib ran

    DT_DOUBLE meanCalibSkinSpec[DT_MAX_SPECTRA_CHANNELS];
s */                                /* mean skin, absorbance unit

    DT_DOUBLE leverTol;              /* leverage tolerance */

    DT_DOUBLE lofTol;                /* lack of fit tolerance */

    DT_DOUBLE distTol;               /* Euclidean or Mahalanobis d
istance tolerance
*/

    DT_DOUBLE predSlopeCorrect;      /* predict linear correct, sl
ope */

    DT_DOUBLE predBiasCorrect;       /* predict linear correct, bi
as */

    DT_DOUBLE predLimitLoEndpoint;   /* predict valid range, lo po
int */

    DT_DOUBLE predLimitHiEndpoint;   /* predict valid range, hi po
int */

    DT_DOUBLE alertLimitLoEndpoint; /* Clinical valid range, low
point */

    DT_DOUBLE alertLimitHiEndpoint; /* Clinical valid range, high
point */

    DT_CONTROL_ELEM ctrlRefMatrix;   /* control reference values
*/

    DT_RANK_ELEM skinPredMatrix[DT_MAX_SKIN_PREDICT_RANKS];
list */                             /* skin predict coefficients

    DT_RANK_ELEM loadMatrix[DT_MAX_OUTLIER_RANKS];
ts */                               /* list of outlier coefficient

    DT_DOUBLE svdLoadVector[DT_MAX_SPECTRA_CHANNELS];
/                                  /* SVD loading coefficients */

```

```

}
DT_SKIN_LIBRARY;

#define DT_BIAS_NUM_DELTA 5
typedef struct
{
    DT_INT      count;          /* count of valid delta readi
ngs */
    DT_INT      oldest;         /* index of oldest delta read
ing */
    DT_INT      correction;     /* bias correction to subtrac
t */
    DT_INT      delta[DT_BIAS_NUM_DELTA]; /* delta readings arra
y */
    DT_RTC_TIME lastUpdate;     /* Date & time of the most re
cent successful
or Quality
Reading */
    DT_UWORD    crc_16;         /* cyclic redundancy check fo
r structure */
}
DT_BIAS_CORRECT;

#define DT_HEMO_AVG_NUM 15      /* number of hemocue readings
in running
average */
typedef struct
{
    DT_INT      oldest[BIN_CODE_LAST - 1];
                                /* index of oldest reading */
    DT_INT      hemo_avg[BIN_CODE_LAST - 1][DT_HEMO_AVG_NUM];
                                /* array of readings per bin
code */
    DT_UWORD    crc_16;         /* cyclic redundancy check fo
r structure */
}
DT_HEMO_AVG;

typedef struct
{
    DT_UWORD    data_type;      /* type of data to store */
    DT_UWORD    dest_spect;     /* destination of spectral/abso
rbance */

```

```

    DT_UWORD    dest_read;        /* destination of readings */
    DT_INT      gd_days_rqd;      /* # good days required all mod
es */
    DT_INT      gd_sitng_rqd;     /* # good sittings required all
modes */
    DT_INT      gd_days;          /* # of good days */
    DT_INT      gd_sittings;      /* # of good sittings */
    DT_RTC_TIME lastGoodDay;      /* Date of last good sitting */
    DT_RTC_TIME lastGoodSit;     /* Date of last countable sitti
ng*/
    DT_UWORD    new_user_info;    /* user data is to be updated */
    /
    DT_UWORD    misc_enables;     /* patient is active or waiting
*/
    DT_UWORD    crc_16;          /* cyclic redundancy check for
structure */
}
DT_CONTROL_COUNTERS;

/* Bit defines for misc_enables */
#define USER_ON_HOLD    0x0001    /* Patient is on hold */
/
#define DISP_HEMO_AVG    0x0002    /* Displays hemocue ave
rage screen */
#define DISP_PRED_AVG    0x0004    /* Displays prediction
averages for
measure mode*/
#define DEMO_ENABLE      0x0008    /* Demo mode if set to
1 */

#define DT_FILE_FORMAT_VERSION 0x0200
typedef struct
{
    DT_UWORD    file_version;     /* file format version in h
ex,
                                i.e., 0x1234 is 12.34 */

    DT_UWORD    data_source;      /* 1 = User card / 2 = Tele
medicine */
    DT_USER_ID   user;            /* user identification */
    DT_SKIN_LIBRARY slb;          /* skin library parameters
*/
    DT_UCHAR     fill[916];       /* fill to make Config Data
Line up at 28514
*/
    DT_CONFIG_DATA cfg;           /* configuration parameters

```

```

    */
}
DT_USER_CARD;

#define DT_VALID_CARD_VERSION 0x0200
typedef struct
{
    DT_RTC_TIME      build_time;          /* manuf build date & time
*/

    DT_UCHAR         serial_num[DT_MAX_SIZ_UNIT_SERIAL_NUM +1];
/* Machine identification */
/

    DT_DOUBLE        sensor_coeff[DT_MAX_SENSOR_CHANNELS];
/* sensor scaling coefficie
nts */

    DT_UWORD         sensor_algor[DT_MAX_SENSOR_CHANNELS];
/* sensor scaling algorithm
s */

    DT_OPER_PARMS    operParms;          /* machine operating parame
ters */

    DT_UWORD         val_cardVers;       /* Valid User Card Version
number
in hex, i.e., 0x1234 is
12.34 */
}
DT_MACHINE_INFO;

typedef struct
{
    DT_OBJECT_HEADER objHdr;          /* Standard object header */
/

    DT_RTC_TIME      sittingDate;      /* Date & Time data was sto
red */
}
DT_EOD_DATE;

typedef struct
{
    DT_OBJECT_HEADER objHdr;          /* standard object header, a
lert DT_ObjectID*/

    DT_UDWORD        bico_user_id;     /* patient id corresponding
to this alert */

```

```

    DT_UCHAR        serial_num[DT_MAX_SIZ_UNIT_SERIAL_NUM + 1];
                        /* instrument id correspondi
ng to this alert
*/
    DT_RTC_TIME      date;                /* date and time of alert
*/
    DT_UWORD         mode;                /* instrument operating mode
*/
    DT_INT           stat_code;           /* status code */
}
DT_ALERT;

```

```

/****

```

```

* Array size defines for Telemedicine structures
****/

```

```

#define DT_PHONE_NUMB_SIZE      35
#define DT_MODEM_INIT_SIZE     255
#define DT_IP_ID_SIZE          11
#define DT_BC_ID_SIZE          11
#define DT_IP_ADDR_SIZE        15
#define DT_FILE_NAME_SIZE      45

```

```

typedef struct

```

```

{
    DT_UCHAR        phoneNumber[DT_PHONE_NUMB_SIZE+1];    /* In
ternet Access
Phone Number */
    DT_UCHAR        alt_phoneNumber[DT_PHONE_NUMB_SIZE+1]; /* Al
ternate Internet
Access Phone Number */
    DT_UWORD        maxRetries;                            /* Ma
ximum number of
transfer attempts per day */
    DT_UWORD        maxRetryDays;                          /* Ma
ximum number of
days to try transfers */
    DT_UWORD        dialingStarts;                         /* St
art dialing BICO
at this time of day */
    DT_UWORD        dialingEnds;                           /* Qu
it dialing at this

```

```

time of day if not done */
}
DT_PHONE_DATA;

typedef struct
{
    DT_UWORD      data_bits;          /* Number of bits per chara
cter */
    DT_UWORD      parity;             /* Modem Protocol Parity */
    DT_UWORD      stop_bits;          /* Number of stop bits */
    DT_BOOL       use_flow_ctl;       /* Use Flow Control */
    DT_BOOL       flow_ctl;           /* Kind of Flow Control */
    DT_UCHAR      mdm_init[DT_MODEM_INIT_SIZE+1]; /* Modem I
nitialization
String */
}
DT_MODEM_DATA;

typedef struct
{
    DT_UCHAR      isp_logonid[DT_IP_ID_SIZE+1]; /* Intern
et Access logon
identity */
    DT_UCHAR      isp_password[DT_IP_ID_SIZE+1]; /* Intern
et Access password
*/
    DT_BOOL       server_assign;      /* ISP Se
rver automatically
assigns IP addr */
    DT_UCHAR      isp_UU_enable;      /* ISP Au
thentication
Protocol enable

= none                                **      0

= Password                            **      1
Authentication Protocol

= Challenge                            **      2
Handshake AP                          */
    DT_UCHAR      isp_ip_addr[DT_IP_ADDR_SIZE+1]; /* Used o
nly if ISP doesn't
auto assign */
    DT_BOOL       ip_hdr_compr;       /* Use IP
Header
Compression */
    DT_BOOL       default_gateway;    /* Use de
fault gateway on

```

```

remote control */
    DT_UCHAR        gateway_ipAddr[DT_IP_ADDR_SIZE+1]; /* Used o
nly if gateway IP
address reqd */
    DT_UCHAR        bico_logonid[DT_BC_ID_SIZE+1];      /* BICO A
ccess logon
identity */
    DT_UCHAR        bico_password[DT_BC_ID_SIZE+1];     /* BICO A
ccess password */
    DT_UCHAR        upload_to[DT_FILE_NAME_SIZE+1];     /* Direct
ory to place
patient readings */
    DT_UCHAR        download_from[DT_FILE_NAME_SIZE+1]; /* Direct
ory to get updates
from */
    DT_UCHAR        bicoProxyServer[DT_IP_ADDR_SIZE+1]; /* Used o
nly if Proxy IP
address reqd */
    DT_UCHAR        bicoFtpServer[DT_IP_ADDR_SIZE+1];   /* Used o
nly if FTP IP
address reqd */
    DT_UCHAR        spare_ipAddr[DT_IP_ADDR_SIZE+1];    /* Used o
nly if another IP
address reqd */
}
DT_ACCESS_INFO;

typedef struct
{
    DT_PHONE_DATA    phone_data;      /* Telemedicine dialing inf
ormation */
    DT_MODEM_DATA    modem_data;      /* Modem ASYNC control info
rmation */
    DT_ACCESS_INFO   internet_defn;   /* Internet logon and inter
net protocol data
*/
    DT_UWORD         crc_16;
}
DT_TELEMED;

/****
* This structure is stored in the first sector of the 186 flash
****/
typedef struct
{
    DT_MACHINE_INFO  m_inf;
    DT_TELEMED       tel_med;

```

```

}
DT_186_FLASH_SECTOR_0;

/****
* 186 Flash sector information & Structure offsets in a sector.
* Am29F010A 2X128K Chips = 32K (32768 byte) sectors
* User card data is stored in 186 flash sectors 1,2,3,& 4
****/
#define DT_186_SECTOR_SIZE 32768L
#define DT_186_NUM_SECTORS 8
#define DT_186_TELEMED_ST sizeof(DT_USER_CARD) /* 0
x6FBA size 0x0060
*/
#define DT_186_BIAS_CORRECT_ST 31148 /* 0
x79AC size
0x001A*/
#define DT_186_HEMO_AVG_ST DT_186_BIAS_CORRECT_ST +
sizeof(DT_BIAS_CORRECT)
#define DT_186_CONTROL_COUNTERS_ST 32198 /* 0
x7DC6 size 0x0014
*/

/****
* Telemedicine Download data is stored in 186 flash sectors 5,6,&
7
****/
#define DT_TEL_USERCARD_ST sizeof(DT_OBJECT_HEADER)
/* 0x4 */
#define DT_TEL_TELEMED_ST sizeof(DT_OBJECT_HEADER)
+
sizeof(DT_USER_CARD)

/****
* PCMCIA Flash sector information & Structure sector assignments.
* Am29F040A 1X512K = 64K (65536 byte) sectors
* Except for TELEMED and HEMO_AVG Each structure starts at 0 in i
ts sector.
****/
#define DT_UC_SECTOR_SIZE 65536L
#define DT_UC_NUM_SECTORS 8
#define DT_UC_TELEMED_ST sizeof(DT_USER_CARD)
#define DT_UC_READINGS_ST 1
#define DT_UC_BIAS_CORRECT_ST 6
#define DT_UC_HEMO_AVG_ST DT_UC_BIAS_CORRECT_ST * D
T_UC_SECTOR_SIZE +
sizeof(DT_BIAS_CORRECT)
#define DT_UC_CONTROL_COUNTERS_ST 7

```



```
/**
 * MODEM Flash sector information
 * AM29F016B eight FLASH chips of 2 Mbytes each
 */
#define DT_MD_SECTOR_SIZE          65536L
#define DT_MD_NUM_SECTORS          256

#endif

/**
 * End of file
 */
```