Data structure for various messages

Common parameters used for periodic data as well as alert data
(As per AIS140 specification)

//Macro:

```
#define     IMEI_NO_SIZE                15  //size of IMEI number
#define     VERSION_NO_SIZE             10  //size of version number
#define     VENDOR_ID_SIZE              8  // size of vendor id
#define     HEADER_PACKET_SIZE          3  // size of header packet
#define     START_CHARACTER             $ // symbol for start char
#define     END_CHARACTER               * // symbol for end char
```

//Global Variables:

```
char    header  [HEADER_PACKET_SIZE] ;          //holds the header packet
char    vendor_id  [VENDOR_ID_SIZE] ;           // holds the vendor id
char    firmware_version   [VERSION_NO_SIZE] ; // holds the version no.
char    imei_no  [IMEI_NO_SIZE] ;               // holds the imei  no.
```

HEALTH MONITORING PARAMETER

(As per AIS140 specification)

//Macro:

```
#define     MESSAGE_BUFF_SIZE           100 // size of message buff
```

//Data structures:

```
/*
        This structure can be used to contain all the health info params.
*/
typedef struct message_healthParam
{
  char msgBuff [MESSAGE_BUFF_SIZE];  // message buffer to hold health data
  int   msgLast;                                // used to append next data

} message_healthParam_t;

/*
        Health parameter structure as per AIS 140
*/

typedef struct Health_Parameter
{
        char        *batteryPercent;                //battery percentage
```

```
    char        *lowBattThresholdValue;     //low battery threshold value
    float       memoryPercent;                  //memory percentage
    float       ignOnPacketFreq;            //packet freq in ignition on state
    float       ignOffPacketFreq;           //packet freq in ignition off state
    char        * digitalInputStatus;           //digital input status
    char        * analogInputStatus;            //analog input status

} Health_Parameter_t;
```

PVT_DATA

(As per AIS140 specification)

//Macro:

```
#define     VEHICLE_NO_SIZE                 10 //size of vehicle no.
#define     MESSAGE_BUFF_SIZE               256// size of msg buff
#define     NETWORK_OPEATOR_SIZE            7 //size of network op.
#define     MOBILE_COUNTRY_CODE_SIZE        3 //size of mcc
#define     MOBILE_NETWORK_CODE_SIZE        2 //size of mnc
#define     LOCATION_AREA_CODE_SIZE         4 //size of lac
#define     GSM_CELL_ID_SIZE                4 //size of gsm cell id
```

//Global Variables:

```
char        veh_reg_no  [VEHICLE_NO_SIZE] ; // holds vehicle no
char        nw_operator [NETWORK_OPEATOR_SIZE] ;//holds n/w op size
char        mcc [MOBILE_COUNTRY_CODE_SIZE] ;  //holds mcc value
char        mnc [MOBILE_NETWORK_CODE_SIZE];  //holds mnc value
char        lac  [LOCATION_AREA_CODE_SIZE];   //holds lac value
char        cell_id  [GSM_CELL_ID_SIZE];       //holds cell id
```

//Data structure:

```
/*
    Enum to determine packet status ,
    required for alert data as well
*/

typedef enum PacketStatus
{
    LIVE_PACKET = 1,            //live packet
    HISTORY_PACKET              //history packet
```

```
} PacketStatus;

/*
	This structure can be used to contain all the PVT params.
*/
typedef struct message_pvt
{
	char msgBuff[256];	// message buffer to hold health data
	int msgLast;		// used to append next data
} message_pvt_t;

/*
	Enum for alert message type as per AIS 140 specification,
	required for alert data as well.
*/
typedef enum Alert_Message
{
	LOCATION_UPDATE			= 0x01,	//1
	LOCATION_UPDATEHISTORY		= 0X02,	//2
	ALERT_BATTERYDISCONNECT		= 0X03,	//3
	ALERT_LOWBATTERY		= 0X04,	//4
	ALERT_LOWBATTERYREMOVED		= 0X05,	//5
	ALERT_CONNECTEDTOBATTERY	= 0X06,	//6
	ALERT_IGNITION_ON		= 0X07,	//7
	ALERT_IGNITION_OFF		= 0X08,	//8
	ALERT_GPS_BOXOPENED		= 0X09,	//9
	ALERT_EMERGENCYSTATEON		= 0X0A,	//10
	ALERT_EMERGENCYSTATEOFF		= 0X0B,	//11
	ALERT_OTA_PARAMCHANGED		= 0X0C,	//12
	ALERT_HARSHBRAKING		= 0X0D,	//13
	ALERT_HARSHACCELERATION		= 0X0E,	//14
	ALERT_RASHTURNING		= 0X0F,	//15
	ALERT_DEVICETAMPERED		= 0X10,	//16

} Alert_Message;

/*
	Structure for parameters of PVT data as per AIS 140 specification
*/
typedef struct PVT_Data
{
	Alert_Message	alertType;		// alert message type
	PacketStatus	packetType;		// Live or History
	char		* gpsFix;		// GPS fix availability
```

```
    char        * date;             //date
    char        * time;             //time
    float       latitude;           //latitude
    char        * latDirection;     //latitude direction
    float       longitude;          //longitude
    char        * longDirection;    //longitute direction
    float       speed;              //speed
    float       heading;            //course over ground in degress
    char        * numSatellites;    //no. of satellites available for fix
    float       altitude;           // altitude of device
    double      pdop;               //positional dilution of precision
    double      hdop;               //horizontal dilution of precision
    char        * ignition;         //Ignition status
    char        * mainPowerStatus;  //Veh. battery connection status
    float       mainInputVoltage;   //source voltage indicator
    float       intBatteryVoltage;  //internal battery voltage.
    char        * emergencyStatus;  //emergency status
    char        * tamperAlert;      //tamper alert
    char        * gsmStrength;      //gsm strength range
    char        * mnc;              //mobile country code
    char        * lac;              //location area code
    char        * cellid;           //GSM cell ID
    char        * nmr;              //network measurement report
    char        * digitalInputStatus;   //digital input status
    char        * digitalOutputStatus;  //digital output status
    char        * frameNumber;      //message sequence
    char        * checksum;         //ensures no error in transmission

} PVT_Data_t;
```

ALERT_MESSAGE

(As per AIS140 specification)

//Macro:

```
# define        MESSAGE_BUFF_SIZE           150 // size of message buff
```

//Global Variables:

```
char            veh_reg_no  [VEHICLE_NO_SIZE] ;     // holds vehicle no
```

// Data structure :

```
/*
        enum to determine GPS quality as per AIS140 specification
*/
typedef enum  Quality
{
        FINE_GPS =  1,                  //gps quality is fine
        COARSE_GPS                      //gps quality is coarse
}Quality;

/*
        This structure can be used to contain all the alert params.
*/
typedef struct message_alert
{
        char   msgBuff[150];      // message buffer to hold health data
        int      msgLast;         // used to append next data
} message_alert_t;

/*
        Structure for parameters of alert message as per AIS 140 specifications
*/
typedef struct Alert_data
{
        char        *packetHeader;          //      3 bytes
        char        *alertType;             //      EMR or SEM
        char        *packetStatus;          //      Normal or Stored
        char        *date;                  //      date
        char        *gpsValidity;           //      gps Validity
        float       lattitude;              //      latitude
        char        *latDirection;          //      latitute direction
        float       longitude;              //      longitude
        char        *longDirection;         //      longitude direction
        float       altitude;               //      altitude
        float       speed;                  //      speed
        float       distance;               //      distance
        char        *gpsQuality;            //      gps quality(G or N)
        char        *replyNo;               //      mobile number
        char        *checksum;              //      checksum number
    } Alert_data_t;
```