

DSHGEM-LIB 通信エンジンライブラリ(GEM+GEM300)
ソフトウェア・パッケージ

変数リミット監視機能 説明書

2011年4月

株式会社データマップ

【取り扱い注意】

- ・ この資料ならびにソフトウェアの一部または全部を無断で使用、複製することはできません。
- ・ 本説明書に記述されている内容は予告なしで変更される可能性があります。
- ・ Windows は米国 Microsoft Corporation の登録商標です。
- ・ ユーザーが本ソフトウェアの使用によって生じた遺失履歴、(株)データマップの予見の有無を問わず発生した特別損害、付随的損害、間接損害およびその他の拡大損害に対して責任を負いません。

【改訂履歴】

番号	改訂日付	項目	概略
1.	2008.1	初版	
2.	2011.4	GenClass クラスライブラリの 項の説明追加	2.2 DSHGenClass クラス・ライブラリ使用 を追加
3.			
4.			

目次

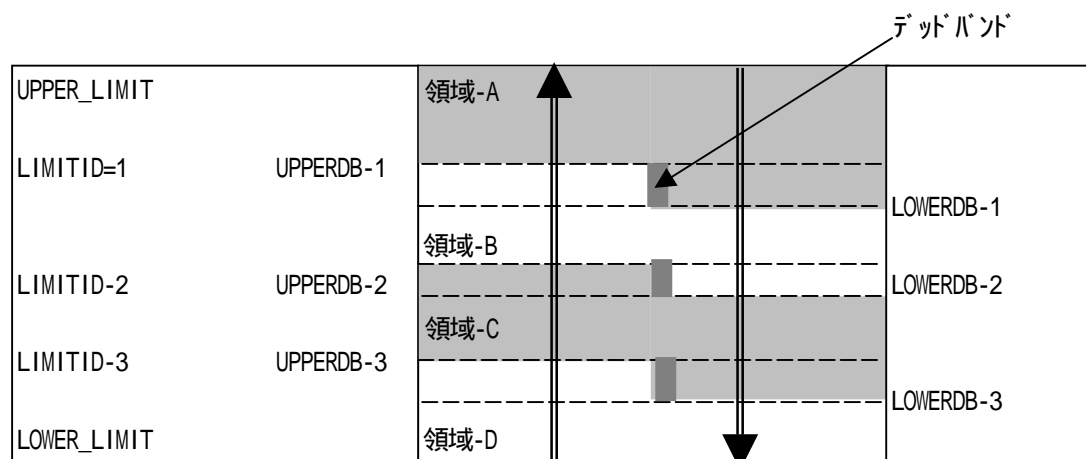
1 . リミット監視の仕様	1
2 . 関連情報	2
2 . 1 DSHGEMLIBエンジン API呼び出しを使用	2
2 . 2 DSHGemClassクラス・ライブラリ使用	4
3 . リミット監視イベントのアプリケーションへの通知機能	5
4 . 処理フローチャート	8

1. リミット監視の仕様

監視対象の変数の値が、ある領域から別の領域へ遷移する際に、領域の境界に UPPERDB と LOWERDB で与えられたデッドバンドを抜け出したときにイベントを通知し、ホストに伝える機能が変数リミット監視機能です。

変数値の領域は、最大8つに分割できます。これにより設けることができるデッドバンドは最大7つになります。

下の図は、4つの領域に分割されたケースです。(3つのLIMITIDが存在します。)



例えば、領域-A について説明すると、UPPERDB-1 と LOWERDB-1 で挟まれた部分が領域-A と他の領域間を遷移する際のデッドバンドになります。そして、次のような変数値の変化による遷移でイベントを発生させます。

変数値が 領域-A 以外の領域から UPPERDB-1 の値以上になり、領域-A に遷移したときにイベントを発生させます。

変数値が領域-A から LOWERDB-1 の値以下になり、他の領域に遷移したときにイベントを発生させます。

UPPER_LIMIT、LOWER_LIMIT は、それぞれ、その変数に与えられた MAX、MIN の値と同じです。

領域-B,C,D についても領域-A と同様になります。

また、DSHGEM-LIB では、リミット監視イベントをホストに通知するだけでなく、DSHGEM-LIB のアプリケーションプログラムに対しても通知する機能を提供します。

2 . 関連情報

2 . 1 DSHGEMLIB エンジン API 呼び出しを使用

変数リミット監視機能に関する情報を整理します。

(1) リミット監視条件の設定

ホストから S2F45 で変数 ID, リミット ID とそのデッドバンドの境界値が与えられる。リミット ID は 1 個以上最大、 7 個まで与えられる。

LIMIT-ID-1、 lowerdb, upperdb 値
-2
.
.
.
-7

APP では、自身で SetMultiVLimit() または、 GemSetVLimit() 関数を使って設定することができます。使用する構造体は次のとおり。

```
typedef struct{
    TLIMITID    limit_id;
    void        *upperdb;
    void        *lowerdb;
} TLIMIT_ID_INFO;

typedef struct{
    TDVID        vid;
    int          limit_count;
    TLIMIT_ID_INFO **limitid_list;
    int          format;           // for upperdb & lowerdb
    int          asize;           // " " " "
} TLIMIT_INFO;
```

(2) 関連する予約 CEID と状態変数(SV)

CEID(イベント ID) : リミット領域遷移時に通知する予約 CEID を設ける。

```
#define CEX_RSV_LIMIT          2
```

GemSetReservedCeid() 関数を使って通知したい CEID を登録する。

SVID(状態変数) : リミット関連の予約装置状態変数 SVID を設ける。
リミット監視によって領域遷移が認められたときに、監視結果情報を設定する状態変数です。

```
#define SVX_RSV_LIMIT_VID      7          // EQ limit ck vid
```

```
#define SVX_RSV_LIMIT_DWVAL      8          // EQ limit variable value
#define SVX_RSV_LIMIT_ID        9          // EQ limit_id
#define SVX_RSV_LIMIT_DIR      10         // EQ limit change direction
```

GemSetReservedSvid () 関数を使って予約 SVID を登録する。

(3) リミット監視結果情報

結果情報は DSHGEM-LIB から APP に次の構造体にセットされて渡される。

```
typedef struct{
    TVID   vid;                // vid
    char   *value;            // value in ascii
    int    limitid;           // limit id
    int    dir;               // transient direction (0=up, 1=down)
}TVLIMIT_EVENT_INFO;
```

(4) 関連関数

使用できる関連関数を示します。

#	関数名	機能
1	API int APIX GemSetMultiVLimit(int eqid, TLIMIT_LIST *lmlist); API int APIX GemSetVLimit(int eqid, TDVID vid, TLIMIT_INFO *lminfo);	複数 VID のリミット設定 1個に VID のリミット設定
2	API int APIX GemSetupVLimitEventList(int eqid, HANDLE event); API void APIX GemTerminateVLimitEventList(int eqid); API int APIX GemPutVLimitEventList(int eqid, TVLIMIT_EVENT_INFO *info); API int APIX GemGetVLimitEventList(int eqid, TVLIMIT_EVENT_INFO *info);	キューの作成とイベント登録 (未使用) 監視結果情報キュー-PUT 監視結果情報キュー-GET
3	API int APIX GemSetReservedCeid(int eqid, int index, TCEID ceid);	予約収集イベントの登録
4	API int APIX GemSetReservedSvid(int eqid, int index, TSVID svid); API int APIX GemGetReservedSvid(int eqid, int index, TSVID *svid);	予約装置状態変数の登録
5	API HANDLE APIX DshCreateThread(LPTHREAD_START_ROUTINE th_func, int para, LPDWORD thid);	スレッド生成関数
6	API void APIX DshFreeTVLIMIT_EVENT_INFO(TVLIMIT_EVENT_INFO *info);	TVLIMIT_EVENT_INFO 内メモリ開放
7	API void APIX DshFreeTLIMIT_INFO(TLIMIT_INFO *info);	TLIMIT_INFO 内メモリ開放

2.2 DSHGemClass クラス・ライブラリ使用

DshLimit クラスを使用します。

- (1) エンジン開始時に予め以下の SV 予約変数設定を行う必要があります。
2.1 と同様に、DshEquipment クラスのメソッドを使って行います。
- (2) 変数リミット監視の情報は次の 2 つの方法によって設定し、監視を開始することができます。

a. ホストからの S2F45 を受信による設定です。DshLimitList クラスを使用します。

使用クラス	メソッド	処理
DshLimitList	decode()	受信した S2F45 に含まれる情報を DshLimitList クラス内にデコードします。
	set()	DshLimitList にデコードしたリミット情報を DSHEng4 エンジンに設定し、リミット監視を開始または停止します。

b. 装置側の APP プログラムで設定です。DshLimit クラスを使用します。

使用クラス	メソッド	処理
DshLimit	set_id()	変数 ID をセットします。 変数情報も DshLimit 内に get します。
	clear()	Class 内のリミット設定情報を消去します。
	add_limitid()	リミット ID と upperdb, lowerdb をクラス内に設定します。
	set()	クラス内の limit 情報をエンジンに設定します。 これでリミット監視が開始します。
	delete()	エンジン内の変数 ID に設定されているリミット情報を削除します。 これで、リミット監視が停止します。

- (3) リミット監視の結果、バンド境界を越えた場合、ホストに対し S6F11 を送信します。
また、それとは別に、APP に対してもイベントを通知してもらうようにすることができます。
通知は、次の DshLimit の static のメソッドを使って予め、APP がイベント通知を受け取るためのハンドラーを設定しておきます。

使用クラス	メソッド	処理
DshLimit	start_event_monitor()	イベントハンドラーを設定し、通知を受取を有効にします。 DshLimit.start_event_monitor()
	stop_event_monitor()	イベントハンドラーへの通知を無効にします。
	get_event_monitor_state()	イベントハンドラーへの通知が有効になっているかどうかの状態を取得します。

関連クラスについての詳細情報は、次の資料を参照ください。

文書番号 DSHGEM-07-30361-00-ClassLib-Info-1

「クラス・ライブラリ説明書 Vol-1」の 5. 変数リミット監視関連クラス

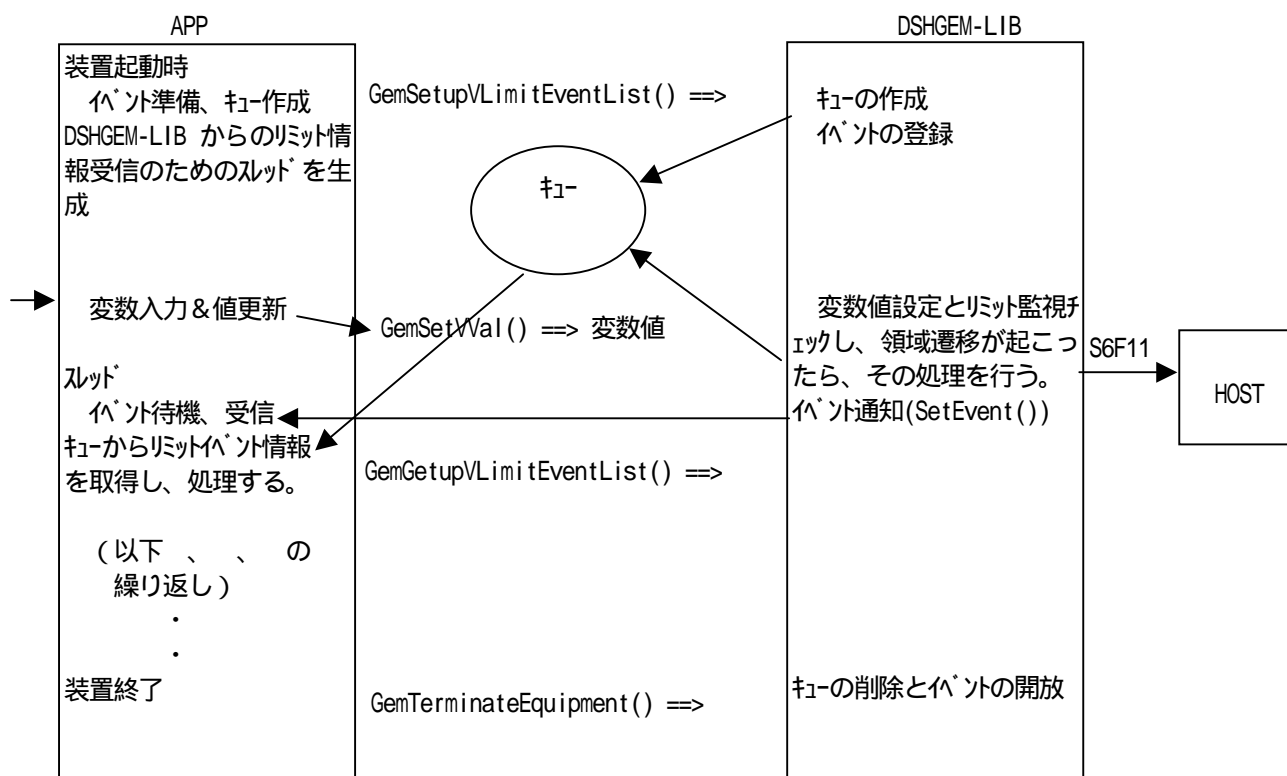
また、詳しいコーディング例は、GemCsDemo デモプログラムの formLimit.cs を参考にしてください。

3 . リミット監視イベントのアプリケーションへの通知機能

3 . の処理で、S6F11 によるリミットイベント通知はホスト側に対して行われますが、アプリケーション側にもリミットのイベント通知を行うための機能です。

DSHGEM-LIB から APP へのリミット監視結果イベントの通知専用キューと Windows のイベント通知機能を使って行います。

以下、制御の流れと概略をチャートで示します。 中の数字の順に進みます。



以下、処理内容を詳しく説明します。

(1) DSHGEM-LIB から APP へのイベント情報を渡す方法

専用キューと Windows のイベント通知機能を使って行います。

キューを通して渡される情報は、変数 ID(vid)、変数値(value)、リミット ID(limitid)と遷移方向(dir)であり、これらを TVLIMIT_EVENT_INFO 構造体に詰めて渡されます。

(2) APP と DSHGEM-LIB の初期準備処理

変数リミットのために使用する予約収集イベント ID (CEID) と状態変数 (SVID) を登録します。この登録は、GemSetupEquipment()関数の直後に行う必要があります。

CEX_RSV_LIMIT - リミット領域遷移通知用イベント

SVX_RSV_LIMIT_V - リミット監視対象変数 ID

SVX_RSV_LIMIT_DWVAL - 同変数値(文字列)

SVX_RSV_LIMIT_ID - 同リミット ID

SVX_RSV_LIMIT_DIR - 同遷移方向

それぞれ、GemSetReservedCeid() と GemSetReservedSvid() 関数を使って予約登録します。

APP が API 関数を使ってイベント情報を受け取るためのキューの準備を要求します。

```
int GemSetupVLimitEventList( int eqid, HANDLE event );
```

event は、CreateEvent() Windows 関数で取得したものを指定します。キューは、DSHGEM-LIB が内部に生成します。TDshList 構造体のものです。

event については、イベント通知を使用しないで、ループの中でポーリングしながら処理したい場合は値を=NULL にしてください。

APP は、DSHGEM-LIB からリミット監視イベント情報を受け取り、処理するためのスレッドを作成します。スレッドにする理由は、APP が他のプログラムと独立的に非同期に受信処理できるようにするためです。スレッドの生成には、DshCreateThread()関数が準備されています。

(3) 変数値更新と DSHGEM-LIB のリミット監視処理

APP が変数の値を API 関数で更新したときに、DSHGEM-LIB リミット監視処理を行います。

値を更新する変数に対するリミット監視が Enable (=リミット監視の設定が行われている) 状態であれば、値が領域遷移を起こしたかどうかを調べます。調べた結果、遷移があれば以降の処理を行います。

変数リミットに予約された CE イベントを S6F11 メッセージでホストに通知します。

変数 ID(vid)、変数値(value)、リミット ID(limitid)と遷移方向(dir)について、予約された状態変数にこれらリミット監視結果情報を設定した後に S6F11 を通知します。

DSHGEM-LIB は監視結果情報を TVLIMIT_EVENT_INFO 構造体に詰めてキューイングし、APP にも通知します。

(1) で準備されたキューにです。使用する関数は次の関数です。

```
int GemPutVLimitEventList( int eqid, TVLIMIT_EVENT_INFO *info );
```

また同時に、event が指定されていれば、APP に対しイベント通知をおこないます。

(4) スレッドによるリミット監視結果情報取得と処理

APP では、(1) で生成したスレッドがイベント待機またはキューのポーリングによって監視します。スレッドは、もし、GemSetupVLimitEventList() 関数でイベントを指定した場合には、イベントを WaitForSingleObject() 関数を使って待機します。また、イベントを指定しなかった場合には周期的にキューから情報取得関数を使って情報があるかどうかを調べ、あれば取得して処理することになります。リミット監視結果情報を取得するための関数は次の通りです。

```
int GemGetVLimitEventList( int eqid, TVLIMIT_EVENT_INFO *info );
```

この関数の戻り値が > 0 ならばイベント情報があったことを意味し、info で指定された構造体にリミット監視結果情報が渡されます。

APP はキューから取得したリミット監視結果情報を解析し処理することになります。処理内容はシステム仕様に基づいて行われることになります。例えば、アラーム通知が必要であれば、S5F1 を送信することになります。

次に、info で指定された TVLIMIT_EVENT_INFO 構造体内の使用メモリを開放します。次の関数を使用します。

```
void DshFreeTVLIMIT_EVENT_INFO( TVLIMIT_EVENT_INFO* info );
```

(5) リミット監視終了処理

(1) で準備したキューの削除とイベントの開放は装置制御の終了時、すなわち、GemTerminateEquipment() 関数が実行されたときに、DSHGEM-LIB が行います。APP 側で行う必要はありません。

4. 処理フローチャート

データ変数の値が更新され、前回の値から変化があったときにリミット監視処理を行います。

