DSHEng5　装置通信エンジン(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

２００９年7月

株式会社データマップ

文書番号　DSHEng5-19-30310-00

［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.	2019.07.08	初版	
2.			
3.			
4.			

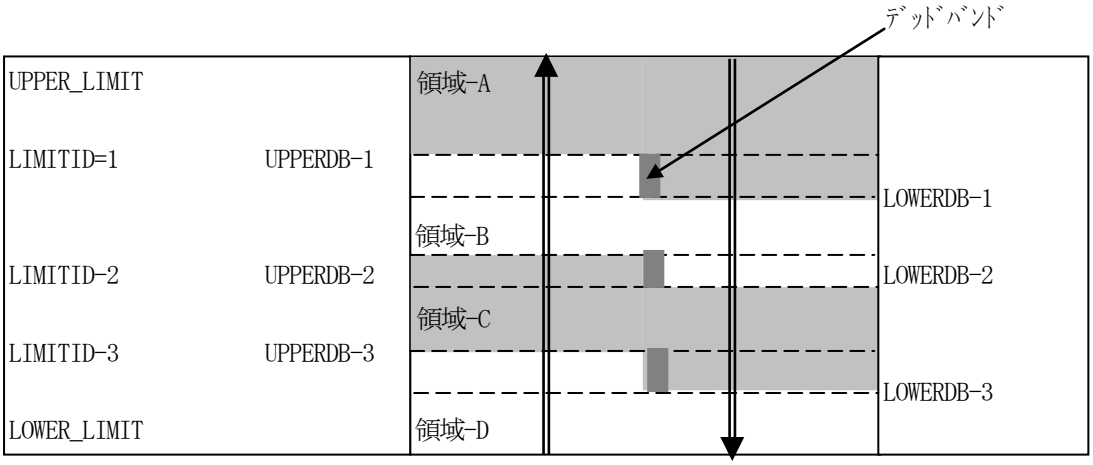
目　次

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、領域の境界として UPPERDB と LOWERDB で与えられたデッドバンドを抜け出したときに、予め指定されているイベントをホストに通知伝達する機能が、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。

①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。

②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_LIMT 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、DSHEng5 では、**リミット監視イベント**をホストに通知するだけではなく、**DSHEng5 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．リミット情報保存クラス

リミット情報は、装置変数（EC, SV, DV）情報保存クラスである TV_INFO のプロパティの中に保存されます。

プロパティ　：　public TLIMIT_INFO limit

２．１　TLIMIT_INFO - 装置変数リミット情報保存クラス

装置変数のリミット情報を保存クラスです。

２．１．１　コンストラクタ

TLIMIT_INFO クラスのインスタンスを生成します。

例 ： V_123 の変数 ID のインスタンスを生成します。

```
UInt32 V_123 = 100;
TLIMIT_INFO V_123 = new TLIMIT_INFO();
```

２．１．２　プロパティ

下表のプロパティを所有しています。

	プロパティ名	説明
1	public UInt32 vid	変数 ID です。(EC, SV, or DV)
2	public int format	変数値のフォーマットです。 B, A, I1, I1...など
3	public int asize	変数値のサイズ
4	lmt_state	リミット監視の状態です。 不定、確定、状態遷移
5	public int c_limitid	(内部処理用)
6	public int lmt_dir	(内部処理用)
7	public int limitid_count	リミット ID 情報配列リストに保存されている limit id 数 (limitid, lowerdb, upperdb)
8	public TLIMIT_ID_INFO[] limitid_list	リミット ID 情報配列リスト TLIMIT_ID_INFO クラスのインスタンスリスト

２．１．３　メソッド

変数リミット情報保存クラスのメソッドは下表のとおりです。

	メソッド名	説明
1	public void Dispose()	インスタンスを破棄します。
2	public void clear()	すべてのプロパティの値をクリアします。
3	public int set_vid()	変数 ID です。
4	public int add_limitid()	limitid_list[]配列リストにリミット ID 情報を追加します。
5	public int set_limitid()	limitid_list[]配列リストの配列位置を指定してリミット ID 情報を設定します。
6	public int copy()	TLIMIT_INFO クラスのインスタンスをコピーします。

２．２　TLIMIT_ID_INFO - リミット ID 情報クラス

リミット ID 情報を保存するクラスです。

２．２．１　コンストラクタ

TLIMIT_ID_INFO クラスのインスタンスを生成します。

２．２．２　プロパティ

下表のプロパティを所有しています。

	プロパティ名	説明
1	public int limit_id	リミット ID
2	public IntPtr upperdb	上限値
3	public IntPtr lowerdb	下限値

２．２．３　メソッド

変数リミット情報保存クラスのメソッドは下表のとおりです。

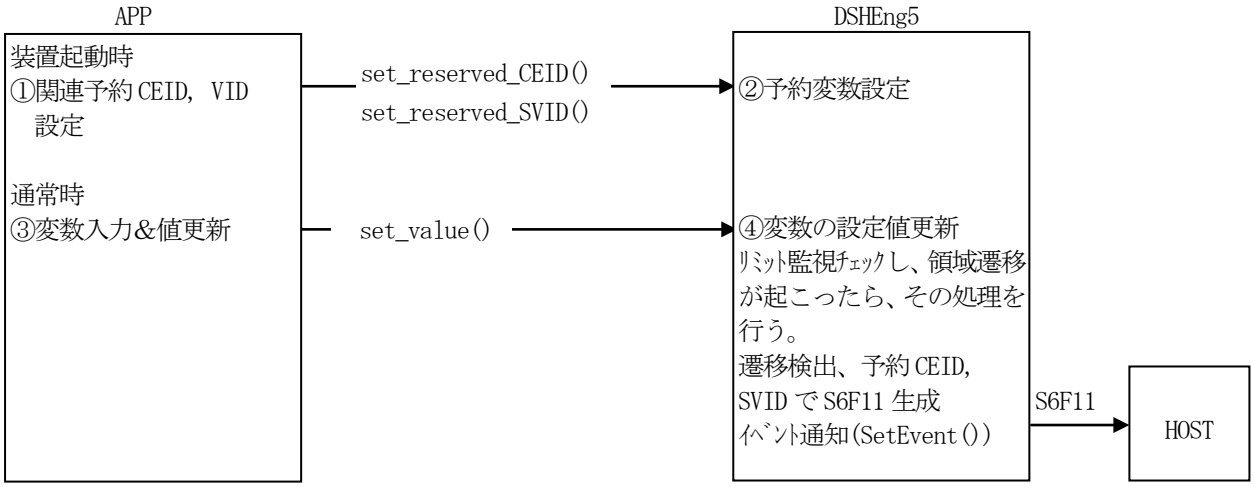
	メソッド名	説明
1	public void Dispose()	インスタンスを破棄します。
2	public void clear()	すべてのプロパティの値をクリアします。
3	public int set()	リミット ID、上限値、下限値を設定します。
4	public static int copy( )	TLIMIT_ID_INFO クラスのインスタンスをコピーします。

３．リミット遷移検出時の DSHEng5 の処理

以下の２種類の処理を行うことができます。

（1）ホストへの S6F11 によるリミット遷移（領域超過）の通知

（2）APP へのイベント通知（イベントハンドラーを呼び出す。



リミットチェック全体の流れについては４．の処理フローチャートを参照ください。

以下、処理内容を詳しく説明します。

３．１　ホストへの S6F11 収集イベント通知について

３．１．１　APP の準備

（1）通知 CEID の設定
APP は、予め、リミット遷移イベント通知を行うための CEID を予め設定しておく必要があります。
設定方法は以下の例のように行います。

```
uint CE_LimitChange =  222;                                          // CEID = 222
class_Reserved_V. set_reserved_CEID(class_const.CEX_RSV_LIMIT, CE_LimitChange);
```

（2）リンク関連変数 SVID の予約

対象変数 ID,　遷移方向、変数値(文字列)、リミット ID の SVID の予約を行います。

```
class_Reserved_V. set_reserved_SVID(class_const. SVX_RSV_LIMIT_VID, SV_LimitVid);
class_Reserved_V. set_reserved_SVID(class_const. SVX_RSV_LIMIT_DIR, SV_LimitDir);
class_Reserved_V. set_reserved_SVID(class_const. SVX_RSV_LIMIT_DVVAL, SV_LimitDVVal);
class_Reserved_V. set_reserved_SVID(class_const. SVX_RSV_LIMIT_ID, SV_Limitid);
```

（3）通知 CEID にリンクするレポート ID の定義、それにリンクする変数 ID の定義
DSHEng5 の標準の定義例は以下の通りです。(デモプログラムで使用する EQINF0.txt 変数定義ファイルのもの、SV_変数は定義されているものとします。)

```
def_ce  CE_LimitChange{                          // CEID の定義
        ceid:     15862                          // =0x00003df6
        enabled:  1
        rptname:  "RP_LimitChange"               // rptid=15872
        }
def_report RP_LimitChange{                       // ReportID
        rptid:     15872                         // =0x00003e00
        vname:    "SV_Clock"                     // vid=検出時刻時刻
        vname:    "SV_LimitVid"                  // vid=対象 VID
        vname:    "SV_LimitDVVal"                // vid=VID 値
        vname:    "SV_LimitLimitid"              // vid=リミット ID (検出した)
        vname:    "SV_LimitDir"                  // vid=値の上昇/下降の識別
        }
```

３．１．２　S6F11 の送信

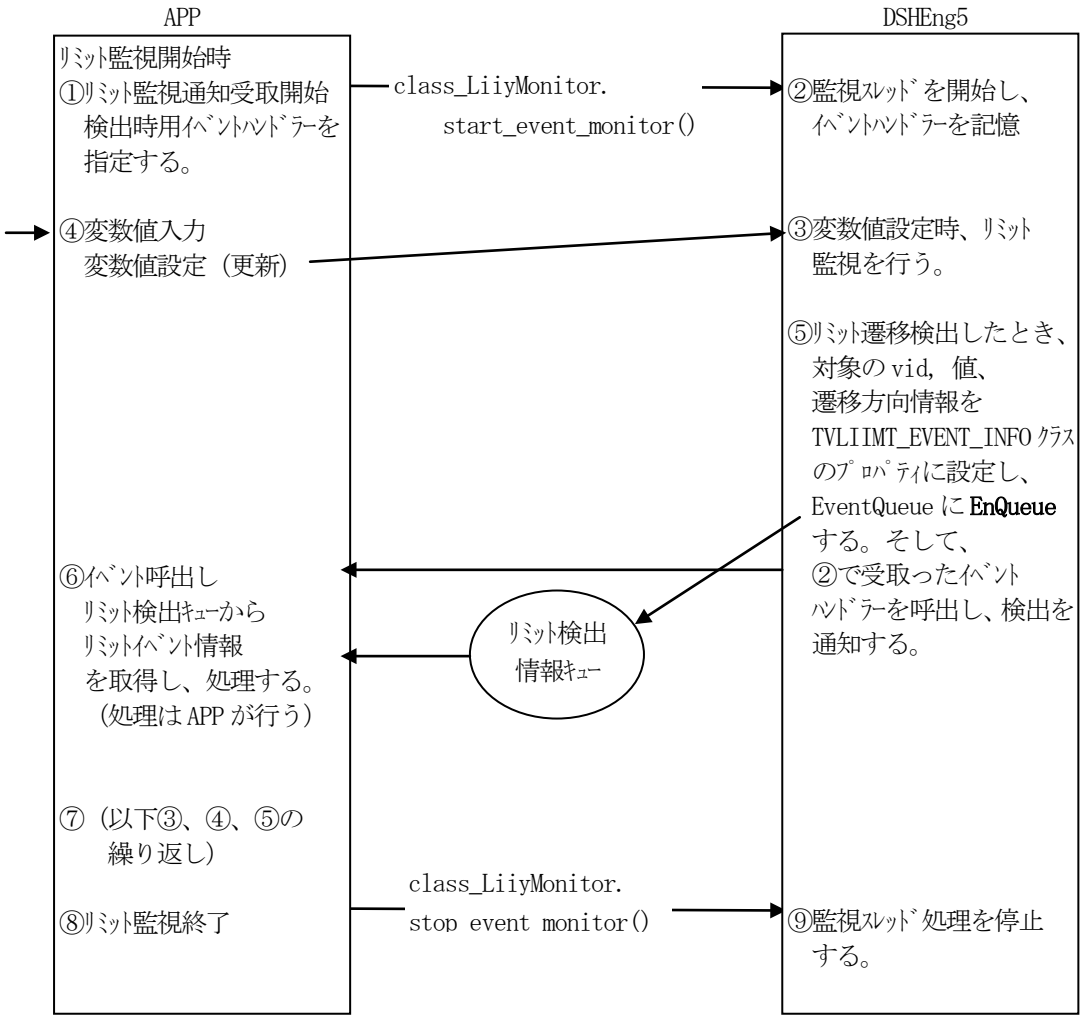
S6F11 の送信は、DSHEng5 が自動的に行います。

３．２　リミット超過検出の APP への通知について

３．１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リミットのイベント通知を行うための機能です。

DSHEng5 から APP へのリミット監視結果イベントの通知を行うための機能です。

以下、制御の流れと概略をチャートで示します。〇の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。

３．２．１　APP 側の準備、検出情報の取得

（1）通知 CEID の設定
　　３．１．１－（1）と同じです。

（2）リンク関連変数 SVID の予約
　　３．１．１－（2）と同じです。

（3）通知 CEID にリンクするレポート ID の定義、それにリンクする変数 ID の定義
　　３．１．１－（1）と同じです。

（4）リミット検出通知を受けるためのイベントハンドラーの DSHEng5 への登録

APP は、通知コールして欲しいイベントハンドラーを次のクラスのメソッドを使って登録します。

public static void start_event_monitor(class_CALLBACK.LimitEventHandler handler)

handler ： ハンドラーのエントリであり、その書式は次の通りです。

public delegate void LimitEventHandler(uint vid, string value, int direction);

vid ：	検出した変数 ID
value:	検出時の変数値（文字列で表現）
direction:	検出方向(upper /lower)

なお、APP では、上記４つの処理は DSHEng5 起動後、１回だけ実行してください。

３．２．２　APP によるリミット検出詳細情報の取得

イベント通知の後、APP はリミット検出情報を DSHEng5 から受取ることができます。（必ず受取ってください）

APP は class__LimitMonitor クラスの get_monitor_queue()メソッドを使って、リミット検出時の情報を取得します。
メソッドの書式は以下の通りです。戻り値が TVLIMIT_EVENT_INFO のインスタンスです。

```
public static TVLIMIT_EVENT_INFO get_req_queue()
```

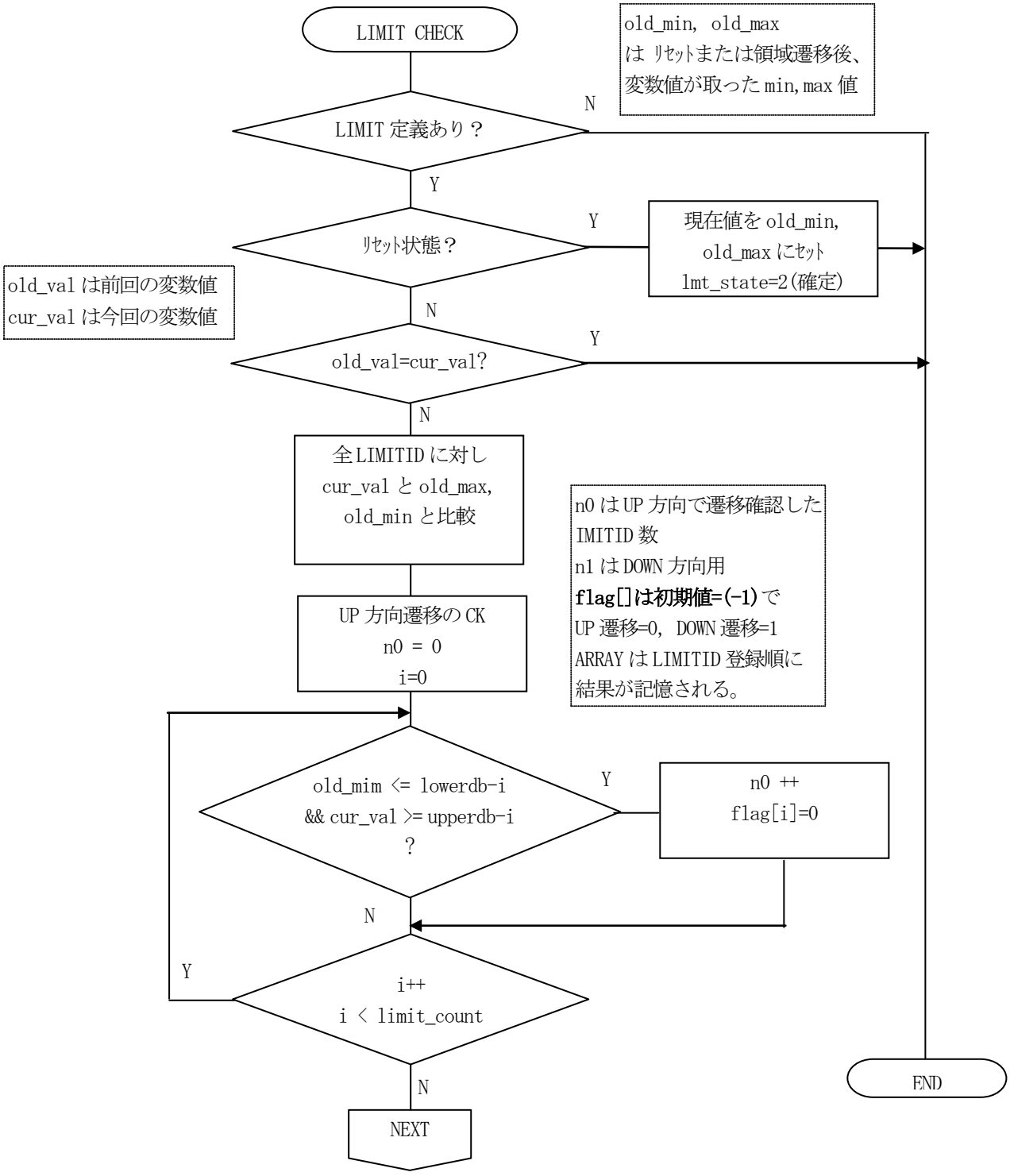
TVLIMIT_EVENT_INFO クラスに含まれる情報

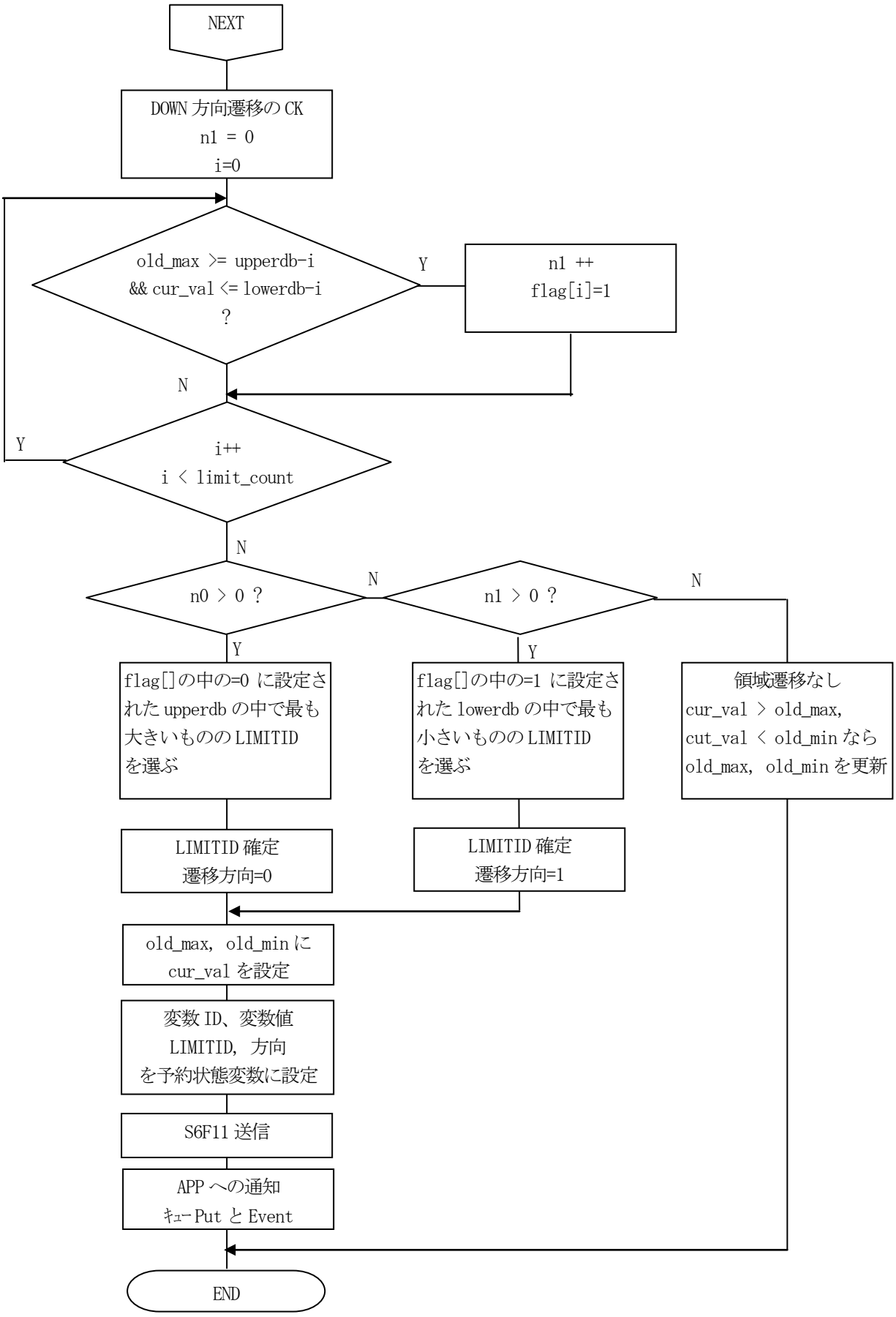
```
public UInt32 vid;                  // vid
public int format;                  // format
public int asize;                   // data size
public string value;                // value in ascii(文字列表現)
public int limitid;                 // limit id
public int dir;                     // transient direction (0=up, 1=down)
```

APP はこれらの詳細情報に従って処理を行うことができます。

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際にリミット監視処理を行います。




NEXT
DOWN 方向遷移の CK
n1 = 0
i=0
old_max >= upperdb-i
&& cur_val <= lowerdb-i
?
Y
n1 ++
flag[i]=1
N
Y
i++
i < limit_count
N
n0 > 0 ?
N
n1 > 0 ?
N
Y
Y
flag[]の中の=0 に設定された upperdb の中で最も大きいものの LIMITID を選ぶ
flag[]の中の=1 に設定された lowerdb の中で最も小さいものの LIMITID を選ぶ
領域遷移なし
cur_val > old_max,
cut_val < old_min なら
old_max, old_min を更新
LIMITID 確定
遷移方向=0
LIMITID 確定
遷移方向=1
old_max, old_min に
cur_val を設定
変数 ID、変数値
LIMITID, 方向
を予約状態変数に設定
S6F11 送信
APP への通知
ｷｭｰPut と Event
END