

GarComm.ocx : Garmin GPS 通信 Active X コントロール リファレンスマニュアル

2002/03/08 Ver 4.1

概要

GarComm.ocx は Garmin プロトコルで Garmin GPS と通信するための Active X コントロールです。

動作環境

Windows 95 以降、Windows NT4.0 以降

動作説明

1. GPS の機種判別

Garmin GPS は機種によって通信フォーマットに違いがあります。フォーマットの識別方法として、どのようなフォーマットを使用しているか GPS に問い合わせる方法と、あらかじめ機種を指定する方法の 2 通りがあり、GarComm.ocx は機種を指定する方法を採用しています。機種を特定するためには GPS の種類毎に割り当てられているプロダクト ID を使います。プロダクト ID を設定するためには、GPS からプロダクト ID を読み出して自動設定する方法と、プログラムから設定する方法があります。

登録されている機種を接続する場合には自動設定できますが、新製品などの登録されていない機種に関しては自動設定できません。もし、登録されていない機種が登録されている機種と同じフォーマットを使用していることがわかれば、登録されている機種のプロダクト ID を設定することで通信することができます。

2. データの読み出しと書き込み動作

GarComm.ocx は GPS から読み出したデータや、GPS に書き込むためのデータを内部バッファに保存します。よって、例えば読み出し動作は、まず GPS から読み出したデータが GarComm.ocx 内部のバッファにコピーされ、GPS からの読み出し後 GarComm.ocx の内部バッファからアプリケーションプログラムに取り出します。

3. 測地系

Garmin プロトコルでは常に WGS84 の測地系でデータの入出力を行います。GarComm.ocx は内部バッファには WGS84 のまま保存します。アプリケーションプログラムに取り出すとき、またアプリケーションプログラムから書き込むときに測地系の変換を行います。

プロパティ

1) long BaudRate	ボーレート
2) short CommPort	COM ポート
3) short Parity	パリティ
4) short DataLength	データ長
5) short StopBit	ストップビット

メソッド

1) GetVersion	GarComm.ocx のバージョンを取得する
2) GetErrorCode	通信エラーの場合のエラーコードを取得する
3) ReadProductID	GPS のプロダクト ID を読み出す
4) SetProductID	プロダクト ID を設定する
5) AbortTransfer	ダウンロード/アップロードを中止する
6) StartNMEAReceiving	NMEA データの受信を開始する
7) StopNMEAReceiving	NMEA データの受信を中止する
8) GetNumSatellite	衛星数を読み出す \$GPGSV
9) GetSatelliteData	PRN 番号で指定した衛星に関するデータを読み出す \$GPGSV

トラック操作

10) DownloadTrack	GPS からトラックデータをダウンロードする
11) GetTrackCount	内部バッファにあるトラック数を取得する
12) GetTrackSegmentCount	トラックセグメント数を取得する
13) GetTrackDataCount	1 つのトラックのデータ数を取得する
14) GetTrackSegmentDataCount	指定したトラックセグメントのデータ数を取得する
15) GetTrackName	トラック名を取得する
16) GetTrackPosition	座標データを取得する
17) GetTrackGarminTime	各データの時刻を 1989/12/31 00:00:00 からの通算秒で取得する
18) UploadTrack	内部バッファにあるトラックデータを GPS にアップロードする
19) AddTrack	アップロードするためのトラック領域を作成する
20) AddTrackSegment	トラックセグメントを追加する
21) AddTrackData	データを追加する
22) DeleteTrackAll	トラック用の内部バッファを開放する

ウェイポイント操作

23) DownloadWaypoint	GPS からウェイポイントデータをダウンロードする
--------------------------------------	---------------------------

	ドする
24) GetWaypointCount	内部バッファにあるウェイポイント数を取得する
25) GetWaypointID	ウェイポイント名称を取得する
26) GetWaypointComment	コメントを取得する
27) GetWaypointLatitude	緯度を取得する
28) GetWaypointLongitude	経度を取得する
29) GetWaypointAltitude	高度を取得する
30) GetWaypointDst	
31) GetWaypointSmb1	シンボル ID を取得する
32) GetWaypointDspl	表示オプションを取得する
33) UploadWaypoint	内部バッファにあるウェイポイントデータを GPS にアップロードする
34) AddWaypoint	ウェイポイントを追加する
35) DeleteWaypointAll	ウェイポイント用の内部バッファを開放する
36) GetWaypointSymbolCount	
37) GetWaypointSymbolName	
38) toWaypointSymbolName	
39) toWaypointSymbolNo	

ルート操作

40) DownloadRoute	GPS からルートデータをダウンロードする
41) GetRouteCount	内部バッファにあるルート数を取得する
42) GetRouteWaypointCount	1 つのルート内のルートウェイポイント数を取得する
43) GetRouteName	ルート名称を取得する
44) GetRouteWaypointLatitude	緯度を取得する
45) GetRouteWaypointLongitude	経度を取得する
46) GetRouteWaypointAltitude	高度を取得する
47) GetRouteWaypointID	ルートウェイポイント名称を取得する
48) GetRouteWaypointComment	コメントを取得する
49) GetRouteWaypointDst	
50) GetRouteWaypointSmb1	シンボル ID を取得する
51) GetRouteWaypointDspl	表示オプションを取得する
52) UploadRoute	内部バッファにあるルートデータを GPS にア

	アップロードする
53) UploadSingleRoute	
54) AddRoute	
55) AddRouteWaypoint	
56) DeleteRouteAll	ルート用の内部バッファを開放する

測地系操作

57) GetDatumCount	サポートしている測地系の数を取得する
58) GetEntryDatumName	測地系名称を取得する
59) GetEntryDatumName2	測地系名称 2 を取得する
60) SelectDatum	測地系を番号で選択する
61) SelectDatumByName	測地系を名称で選択する
62) GetDatumName	現在選択されている測地系の名称を取得する
63) GetSelectedDatumNumber	現在選択されている測地系番号を取得する
64) GetMapDatumDeltaA	現在選択されている測地系の A パラメータを取得する
65) GetMapDatumDeltaF	現在選択されている測地系の F パラメータを取得する
66) GetMapDatumDeltaX	現在選択されている測地系の X パラメータを取得する
67) GetMapDatumDeltaY	現在選択されている測地系の Y パラメータを取得する
68) GetMapDatumDeltaZ	現在選択されている測地系の Z パラメータを取得する
69) TransformDatumWGS84IntoLocal	WGS84 から現在選択されている測地系へ変換する
70) TransformDatumLocalIntoWGS84	現在選択されている測地系から WGS84 へ変換する
71) CalcDistance	2 点間の距離を求める

イベント

```
void FireReceiving(long nReceived, long nTotal)
    {FireEvent(eventidReceiving,EVENT_PARAM(VTS_I4      VTS_I4),
nReceived, nTotal);}
void FireNMEAReceived(short valid, double lat, double lon, double alt, long t,
double vel, short dir)
```

BaudRate プロパティ

ボーレートを設定または取得します。

構文

object.**BaudRate** [= *value*]

BaudRate プロパティの構文の指定項目は次のとおりです。

指定項目	内容
<i>value</i>	ボーレートを指定する整数値

解説

Garmin プロトコルでは 9600、NMEA0183 プロトコルでは 4800 を指定してください。

注意

GPS と通信を行う前に、BaudRate プロパティを設定してください。

データ型

整数型 (long)

CommPort プロパティ

通信ポート番号を設定または取得します。

構文

object.CommPort [= *value*]

CommPort プロパティの構文の指定項目は次のとおりです。

指定項目	内容
<i>value</i>	ポート番号を指定する整数値

解説

GPS との通信に使用する COMM ポート番号を指定してください。

COM1 は 1、COM2 は 2 .. です。

注意

GPS と通信を行う前に、CommPort プロパティを設定してください。

データ型

整数型 (short)

Parity プロパティ

パリティビットを設定または取得します。

構文

object.Parity [= *value*]

Parity プロパティの構文の指定項目は次のとおりです。

指定項目	内容
<i>Value</i>	0~2 = 無し、ODD、EVEN

解説

Garmin プロトコルでは、パリティ無し(0)を指定してください。

注意

GPS と通信を行う前に、Parity プロパティを設定してください。

データ型

整数型 (short)

DataLength プロパティ

データ長を設定または取得します。

構文

object.DataLength [= *value*]

DataLength プロパティの構文の指定項目は次のとおりです。

指定項目	内容
<i>Value</i>	4 ~ 8

解説

Garmin プロトコルでは、8 を指定してください。

注意

GPS と通信を行う前に、DataLength プロパティを設定してください。

データ型

整数型 (short)

StopBit プロパティ

ストップビットを設定または取得します。

構文

object.Parity [= *value*]

StopBit プロパティの構文の指定項目は次のとおりです。

指定項目	内容
<i>Value</i>	0、1、2 = 1、1.5、2

解説

Garmin プロトコルでは、ストップビット 1 (0) を指定してください。

注意

GPS と通信を行う前に、StopBit プロパティを設定してください。

データ型

整数型 (short)

GetVersion メソッド

GarComm.ocx コントロールのバージョンを取得します。

```
long GetVersion(void);
```

パラメータ

無し

戻り値

上位 16 ビットにメジャーバージョン、下位 16 ビットにマイナーバージョンを格納した値が返ります。

解説

動作を保証するために、バージョンのチェックを行ってください。

GetErrorCode メソッド

GPS との通信でエラーが発生した場合のエラーコードを取得します。

```
long GetErrorCode(void);
```

パラメータ

無し

戻り値

winerror.h で定義されるエラーコードが返ります。

解説

通信エラー以外は正しいエラーコードを返しません。

ReadProduct ID メソッド

GPS からプロダクト ID を読み出します。

```
BSTR ReadProductID(long* id, long* ver);
```

パラメータ

id

プロダクト ID。

エラーの場合は、-1 が返ります。

ver

ファームウェアのバージョン番号の 100 倍の値。

戻り値

GPS から取得されるプロダクトとバージョンを示す文字列が返ります。

解説

プロダクト ID 取得後、接続している GPS を GarComm.ocx がサポートしているかどうかを確かめるために、SetProductID メソッドを実行してください。

参照

[SetProductID](#)

SetProductID メソッド

通信する GPS を指定します。

```
BOOL SetProductID(long id);
```

パラメータ

id

GPS のプロダクト ID。

戻り値

id で指定した GPS を GarComm.ocx がサポートしていれば TRUE が、そうでなければ FALSE が返ります。

解説

新機種などの理由でまだ GarComm.ocx がサポートしていない GPS と通信する場合、その GPS がサポート済みの他の機種と同じプロトコルを使っていることがわかればサポート済みの機種を指定することで通信可能です。例えば、eMap、eTrex、eTrex Summit はすべて同じプロトコルです。

AbortTransfer メソッド

GPS との通信を中止します。

```
void AbortTransfer(void);
```

パラメータ

無し

戻り値

無し

解説

Garmin プロトコルでのダウンロード / アップロードを中止します。このメソッドは、ダウンロード、アップロードを別スレッドで実行しているとき、または Receiving イベントの中で使用できます。

StartNMEAReceiving メソッド

NMEA0183 データの受信を開始します。

BOOL StartNMEAReceiving(void);

パラメータ

無し

戻り値

成功すれば TRUE が返ります。通常 FALSE が返ることはありません。

解説

このメソッド実行後、GPS から NMEA0183 のセンテンスを受信すると、FireNMEAReceived イベントが発生します。

参照

FireNMEAReceived

StopNMEAReceiving メソッド

NMEA0183 データの受信を終了します。

```
void StopNMEAReceiving(void);
```

パラメータ

無し

戻り値

無し

解説

このメソッドを実行すると、それ以上 NMEA0183 センテンスの受信を通知しません。

GetNumSatellite メソッド

NMEA0183 のデータ受信中に、補足している衛星数を取得します。

```
long GetNumSatellite(void);
```

パラメータ

無し

戻り値

現在補足中の衛星の個数。

解説

NMEA0183 データによるナビゲーション中のみ有効なメソッドです。

NMEA0183 の\$GPGSV センテンスから補足中の衛星数を取得します。

この情報は\$GPGSV センテンスを受信するたびに更新されるので、このメソッドは NMEAReceived イベント内で使用することを想定しています。

参照

NMEAReceived

GetSatelliteData メソッド

受信中の衛星に関する情報を取得します。

```
BOOL GetSatelliteData(  
    short n,  
    short* PRNnumber,  
    short* ElevationAngle,  
    short* AzimuthAngle,  
    short* SignalStrength  
);
```

パラメータ

n

取得したい衛星を指定します。n は 0 ~ GetNumSatellite()-1 までの間の整数値です。

PRNnumber,

PRN 番号。

ElevationAngle,

衛星を見上げる角度（度）

AzimuthAngle,

衛星の方位角（度）

SignalStrength

信号強度

戻り値

n に範囲外の値を指定すると FALSE が、それ以外は TRUE が返ります。

解説

天空図を表示するときに使用します。

DownloadTrack メソッド

GPS に保存されているトラックログデータを読み出します。読み出されたデータは GarComm.ocx の内部バッファに保存されます。

long DownloadTrack(void);

パラメータ

無し

戻り値

すべてのデータを読み込めれば TRUE を、それ以外は FALSE が返ります。

解説

トラックデータをダウンロードすると、それ以前に内部に保存されていたトラックデータは削除されます。

DownloadTrack を別スレッドで実行している途中で AbortTransfer メソッドを実行すると転送を中断できます。

ダウンロード中には Receiving イベントが発生します。

読み出したデータは、GetTrackCount、GetTrackSegmentCount、GetTrackDataCount、GetTrackSegmentDataCount、GetTrackName、GetTrackPosition、GetTrackGarminTime の各メソッドを使用して内部バッファから取得します。

参照

[AbortTransfer](#)、[Receiving](#)、[GetTrackCount](#)、[GetTrackSegmentCount](#)、[GetTrackDataCount](#)、[GetTrackSegmentDataCount](#)、[GetTrackName](#)、[GetTrackPosition](#)、[GetTrackGarminTime](#)

GetTrackCount メソッド

GPS からダウンロードしたトラックデータのトラック数を取得します。

```
long GetTrackCount(void);
```

パラメータ

無し

戻り値

トラック数

解説

GPS3+、e シリーズなど記録したトラックログをセーブできる機種では、セーブする毎にトラック数が増加します。例えば 10 データまでセーブできる機種ではアクティブログを含めて最大 11 トラックになります。

参照

GetTrackSegmentCount メソッド

トラックセグメント数を取得します。

```
long GetTrackSegmentCount(  
    long tno  
);
```

パラメータ

tno

トラック番号。0 ~ GetTrackCount()-1 を指定します。

戻り値

tno で指定したトラックのトラックセグメント数。

解説

アクティブログは、電源の ON、衛星をロスとした後の再補足でトラックデータのマークが入り、マークから次のマークまでのトラックデータを 1 つのセグメントとして扱います。このメソッドは 1 つのトラックがいくつのセグメントに分かれているかを取得します。

注意

セーブしたトラックは、セグメントの区切りマークが削除されてしまいます。

参照

GetTrackDataCount メソッド

1 つのトラックに含まれるデータ数を取得します。

```
long GetTrackDataCount(  
    long tno  
);
```

パラメータ

tno

トラック番号。0 ~ GetTrackCount()-1 を指定します。

戻り値

指定したトラックのデータ数。

解説

1 つのトラックが複数のセグメントに分けられている場合、各セグメントに含まれるデータ数の合計になります。

参照

GetTrackSegmentDataCount メソッド

1つのトラックセグメントに含まれるデータ数を取得します。

```
long GetTrackSegmentDataCount(  
    long tno,  
    long sno  
);
```

パラメータ

tno

トラック番号。0 ~ GetTrackCount()-1 を指定します。

sno

セグメント番号。0 ~ GetTrackSegmentCount()-1 を指定します。

戻り値

指定したトラックセグメントのデータ数。

解説

参照

GetTrackName メソッド

トラックの名称を取得します。

```
BSTR GetTrackName(  
    long tno  
);
```

パラメータ

tno

トラック番号。0 ~ GetTrackCount()-1 を指定します。

戻り値

トラックの名称が返ります。

解説

参照

GetTrackPosition メソッド

トラックの座標データを取得します。

```
BOOL GetTrackPosition(  
    long tno,  
    long sno,  
    long dno,  
    double* lat,  
    double* lon,  
    double* alt  
);
```

パラメータ

tno

トラック番号。0 ~ GetTrackCount()-1 を指定します。

sno

セグメント番号。0 ~ GetTrackSegmentCount()-1 を指定します。

dno

データ番号。0 ~ GetTrackSegmentDataCount()-1 を指定します。

lat

緯度を返します。南緯は負の値になります。

lon

経度を返します。西経は負の値になります。

alt

高度を返します。

戻り値

成功すれば TRUE を、範囲外の値を指定すると FALSE が返ります。

解説

このメソッドで測地系の変換が行われます。

参照

GetTrackGarminTime メソッド

トラックの時刻データを取得します。

```
long GetTrackGarminTime(  
    long tno,  
    long sno,  
    long dno  
);
```

パラメータ

tno

トラック番号。0 ~ GetTrackCount()-1 を指定します。

sno

セグメント番号。0 ~ GetTrackSegmentCount()-1 を指定します。

dno

データ番号。0 ~ GetTrackSegmentDataCount()-1 を指定します。

戻り値

1989年12月31日 0時0分0秒からの通算秒が返ります。範囲外の値を指定すると0が返ります。

解説

取得した値に 631065600 を加えると、UNIXのCランタイムライブラリで扱う1970年1月1日 0時0分0秒からの通算秒に変換できます。

参照

UploadTrack メソッド

GarComm.ocx の内部バッファに保存されているトラックデータを GPS にアップロードします。

BOOL UploadTrack(void);

パラメータ

無し

戻り値

アップロードに成功すれば TRUE、失敗すれば FALSE が返ります。

解説

UploadTrack は内部バッファの全データを一括してアップロードします。

参照

AddTrack メソッド

GarComm.ocx の内部バッファにトラックを保存する領域を追加します。

```
BOOL AddTrack(  
    BSTR* name  
);
```

パラメータ

name

トラックの名称を指定します。

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

追加されたトラック領域にはデータを保存するセグメントはないので、AddTrack 実行後に AddTrackSegment を実行して、少なくとも 1 つのセグメントを作成してください。

参照

[AddTrackSegment](#)

AddTrackSegment メソッド

指定したトラックにセグメントを追加します。

```
BOOL AddTrackSegment(  
    long tno  
);
```

パラメータ

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

tno で指定したトラックにセグメントを追加します。AddTrack で作成したトラックに対しては、AddTrackSegment で最低 1 つのトラックセグメントを作成してください。

AddTrackData メソッド

トラックデータを追加します。

BOOL AddTrackData(

long tno,

long sno,

double lat,

double lon,

double alt,

long t

);

パラメータ

tno

データを追加するトラック番号を指定します。

sno

データを追加するセグメント番号を指定します。

lat

緯度を指定します。

lon

経度を指定します。

alt

高度を指定します。

t

時刻を 1970 年 1 月 1 日 0 時 0 分 0 秒からの通産秒で指定します。

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

tno と sno で指定したトラックセグメントの最後にトラックデータを追加します。

DeleteTrackAll メソッド

GarComm.ocx のトラックデータ用の内部バッファを開放します。

```
void DeleteTrackAll(void);
```

パラメータ

無し

戻り値

無し

解説

このメソッドを実行すると、GarComm.ocx 内部の全トラックデータは消去されます。

DownloadWaypoint メソッド

GPS に保存されているウェイポイントデータを読み出します。読み出されたデータは GarComm.ocx の内部バッファに保存されます。

long DownloadWaypoint(void);

パラメータ

無し

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

ウェイポイントデータをダウンロードすると、それ以前に内部に保存されていたウェイポイントデータは削除されます。

DownloadWaypoint を別スレッドで実行している途中で AbortTransfer メソッドを実行すると転送を中断できます。

ダウンロード中には Receiving イベントが発生します。

読み出したデータは、GetWaypointCount、GetWaypointID、GetWaypointComment、GetWaypointLatitude、GetWaypointLongitude、GetWaypointAltitude、GetWaypointDst、GetWaypointSmbL、GetWaypointDspl の各メソッドを使用して内部バッファから取得します。

参照

[AbortTransfer](#)、[Receiving](#)、[GetWaypointCount](#)、[GetWaypointID](#)、[GetWaypointComment](#)、[GetWaypointLatitude](#)、[GetWaypointLongitude](#)、[GetWaypointAltitude](#)、[GetWaypointDst](#)、[GetWaypointSmbL](#)、[GetWaypointDspl](#)

GetWaypointCount メソッド

GarComm.ocx の内部バッファに保存されているウェイポイント数を取得します。

```
long GetWaypointCount(void);
```

パラメータ

無し

戻り値

ウェイポイント数が返ります。

解説

参照

GetWaypointID メソッド

ウェイポイント名称を取得します。

```
BSTR GetWaypointID(  
    long no  
);
```

パラメータ

no

ウェイポイント番号。0 ~ GetWaypointCount()-1 までの間の整数値です。

戻り値

成功すればウェイポイント名称、失敗すれば NULL が返ります。

解説

参照

GetWaypointComment メソッド

ウェイポイントのコメントを取得します。

```
BSTR GetWaypointComment(  
    long no  
);
```

パラメータ

no

ウェイポイント番号。0 ~ GetWaypointCount()-1 までの間の整数値です。

戻り値

成功すればウェイポイントのコメント、失敗すれば NULL が返ります。

解説

参照

GetWaypointLatitude メソッド

ウェイポイントの緯度を取得します。

```
double GetWaypointLatitude(  
    long no  
);
```

パラメータ

no

ウェイポイント番号。0 ~ GetWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば緯度、失敗すれば-9999.9 が返ります。

解説

参照

GetWaypointLongitude メソッド

ウェイポイントの経度を取得します。

```
double GetWaypointLongitude(  
    long no  
);
```

パラメータ

no

ウェイポイント番号。0 ~ GetWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば経度、失敗すれば-9999.9 が返ります。

解説

参照

GetWaypointAltitude メソッド

ウェイポイントの高度を取得します。

```
double GetWaypointAltitude(  
    long no  
);
```

パラメータ

no

ウェイポイント番号。0 ~ GetWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば高度、失敗すれば-9999.9 が返ります。

解説

参照

GetWaypointDst メソッド

ウェイポイントの Dst パラメータを取得します。

```
double GetWaypointDst(  
    long no  
);
```

パラメータ

no

ウェイポイント番号。0 ~ GetWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば Dst の値、失敗すれば-1.0 が返ります。

解説

参照

GetWaypointSmbI メソッド

ウェイポイントに指定されているシンボル（アイコン）番号を取得します。

```
short GetWaypointSmbI(  
    long no  
);
```

パラメータ

no

ウェイポイント番号。0 ~ GetWaypointCount()-1 までの間の整数値です。

戻り値

成功すればシンボル番号、失敗すれば0が返ります。

解説

参照

GetWaypointDspl メソッド

ウェイポイントの表示オプションを取得します。

```
short GetWaypointDspl(  
    long no  
);
```

パラメータ

no

ウェイポイント番号。0 ~ GetWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば表示オプション、失敗すれば-1 が返ります。

解説

表示オプションは、GPS の画面上のウェイポイントを表示するときに、シンボル、名称などを表示するかどうかを指定します。この値は GPS モデルによって同じ値で意味が異なります。

参照

UploadWaypoint メソッド

GarComm.ocx の内部バッファに保存されているウェイポイントデータを GPS にアップロードします。

BOOL UploadWaypoint(void);

パラメータ

無し

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

参照

AddWaypoint メソッド

GarComm.ocx の内部バッファにウェイポイントを追加します。

```
BOOL AddWaypoint(  
    BSTR* name,  
    BSTR* comment,  
    double lat,  
    double lon,  
    double alt,  
    short smbl,  
    short dspl,  
    double dst  
);
```

パラメータ

name

ウェイポイント名称を指定します。

comment

ウェイポイントのコメントを指定します。

lat

緯度を指定します。

lon

経度を指定します。

alt

高度を指定します。

smbl

シンボル (アイコン) を指定します。

dspl

表示オプションを指定します。

dst

dst パラメータを指定します。

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

AddWaypoint は同じ名称で既にウェイポイントがあるかどうかの確認は行いません。

参照

DeleteWaypointAll メソッド

GarComm.ocx のウェイポイントデータ用の内部バッファを開放します。

```
void DeleteWaypointAll(void);
```

パラメータ

無し

戻り値

無し

解説

このメソッドを実行すると、GarComm.ocx 内部の全ウェイポイントデータは消去されます。

参照

GetWaypointSymbolCount メソッド

ウェイポイントのシンボル（アイコン）数を取得します。

```
long GetWaypointSymbolCount(void);
```

パラメータ

無し

戻り値

使用できるシンボルの個数が返ります。

解説

GetWaypointSymbolCount は現在設定されているプロダクト ID で使用できるシンボルの個数を取得できます。

参照

GetWaypointSymbolName メソッド

ウェイポイントのシンボル（アイコン）名を取得します。

```
BSTR GetWaypointSymbolName(  
    long no  
);
```

パラメータ

0 ~ GetWaypointSymbolCount()-1 までの間の整数値です。

戻り値

シンボル名が返ります。

解説

参照

toWaypointSymbolName メソッド

Garmin フォーマットで定義されているシンボル(アイコン)番号からシンボルの名称に変換します。

```
BSTR toWaypointSymbolName(  
    long no  
);
```

パラメータ

no

Garmin フォーマットで定義されているシンボル番号を指定します。

戻り値

シンボル名が返ります。

解説

GPS から読み出したシンボル番号から名称を得たいときに使用します。

参照

toWaypointSymbolNo メソッド

ウェイポイントのシンボル（アイコン）名称からシンボル番号に変換します。

```
long toWaypointSymbolNo(  
    BSTR* name  
);
```

パラメータ

シンボル名称を指定します。

戻り値

シンボル番号が返ります。無効な名称を指定すると、18（ウェイポイントドット）が返ります。

解説

GPS へアップロードするときにシンボル名称からシンボル番号に変換するために使用します。

参照

DownloadRoute メソッド

GPS に保存されているルートデータを読み出します。読み出されたデータは GarComm.ocx の内部バッファに保存されます。

long DownloadRoute(void);

パラメータ

無し

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

ルートデータをダウンロードすると、それ以前に内部に保存されていたルートデータは削除されます。

DownloadRoute を別スレッドで実行している途中で AbortTransfer メソッドを実行すると転送を中断できます。

ダウンロード中には Receiving イベントが発生します。

読み出したデータは、GetRouteCount、GetRouteWaypointCount、GetRouteName、GetRouteWaypointLatitude、GetRouteWaypointLongitude、GetRouteWaypointAltitude、GetRouteWaypointID、GetRouteWaypointComment、GetRouteWaypointDst、GetRouteWaypointSmbL、GetRouteWaypointDspl の各メソッドを使用して内部バッファから取得します。

参照

[AbortTransfer](#)、[Receiving](#)、[GetRouteCount](#)、[GetRouteWaypointCount](#)、[GetRouteName](#)、[GetRouteWaypointLatitude](#)、[GetRouteWaypointLongitude](#)、[GetRouteWaypointAltitude](#)、[GetRouteWaypointID](#)、[GetRouteWaypointComment](#)、[GetRouteWaypointDst](#)、[GetRouteWaypointSmbL](#)、[GetRouteWaypointDspl](#)

GetRouteCount メソッド

GarComm.ocx の内部バッファに保存されているルート数を取得します。

```
long GetRouteCount(void);
```

パラメータ

無し

戻り値

ルート数が返ります。

解説

各ルートに設定されているルートウェイポイント数は `GetRouteWaypointCount` で取得します。

参照

[GetRouteWaypointCount](#)

GetRouteWaypointCount メソッド

各ルート毎のウェイポイント数を取得します。

```
long GetRouteWaypointCount(  
    long no  
);
```

パラメータ

no

ルート番号を 0 ~ GetRouteCount()-1 の間で指定します。

戻り値

ルートウェイポイント数が返ります。

解説

GetRouteCount と GetRouteWaypointCount を使用して GPS から読み出したルートに関するデータ数を取得します。

参照

[GetRouteCount](#)

GetRouteName メソッド

ルート名称を取得します。

```
BSTR GetRouteName(  
    long rno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

戻り値

成功すればルート名称、失敗すれば NULL が返ります。

解説

参照

GetRouteWaypointLatitude メソッド

ルートウェイポイントの緯度を取得します。

```
double GetRouteWaypointLatitude(  
    long rno,  
    long wno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

wno

ルートウェイポイント番号。0 ~ GetRouteWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば緯度、失敗すれば-9999.9 が返ります。

解説

参照

GetRouteWaypointLongitude メソッド

ルートウェイポイントの経度を取得します。

```
double GetRouteWaypointLongitude(  
    long rno,  
    long wno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

wno

ルートウェイポイント番号。0 ~ GetRouteWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば経度、失敗すれば-9999.9 が返ります。

解説

参照

GetRouteWaypointAltitude メソッド

ルートウェイポイントの高度を取得します。

```
double GetRouteWaypointAltitude(  
    long rno,  
    long wno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

wno

ルートウェイポイント番号。0 ~ GetRouteWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば高度、失敗すれば-9999.9 が返ります。

解説

参照

GetRouteWaypointID メソッド

ルートウェイポイント名称を取得します。

```
BSTR GetRouteWaypointID(
```

```
    long rno,
```

```
    long wno
```

```
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

wno

ルートウェイポイント番号。0 ~ GetRouteWaypointCount()-1 までの間の整数値です。

戻り値

成功すればルートウェイポイント名称、失敗すれば NULL が返ります。

解説

参照

GetRouteWaypointComment メソッド

ルートウェイポイントのコメントを取得します。

```
BSTR GetRouteWaypointComment(  
    long rno,  
    long wno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

wno

ルートウェイポイント番号。0 ~ GetRouteWaypointCount()-1 までの間の整数値です。

戻り値

成功すればルートウェイポイントのコメント、失敗すれば NULL が返ります。

解説

参照

GetRouteWaypointDst メソッド

ルートウェイポイントの Dst パラメータを取得します。

```
double GetRouteWaypointDst(  
    long rno,  
    long wno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

wno

ルートウェイポイント番号。0 ~ GetRouteWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば Dst の値、失敗すれば-1.0 が返ります。

解説

参照

GetRouteWaypointSmbI メソッド

ルートウェイポイントに指定されているシンボル (アイコン) 番号を取得します。

```
short GetRouteWaypointSmbI(  
    long rno,  
    long wno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

wno

ルートウェイポイント番号。0 ~ GetRouteWaypointCount()-1 までの間の整数値です。

戻り値

成功すればシンボル番号、失敗すれば0が返ります。

解説

参照

GetRouteWaypointDspl メソッド

ルートウェイポイントの表示オプションを取得します。

```
short GetRouteWaypointDspl(  
    long rno,  
    long wno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

wno

ルートウェイポイント番号。0 ~ GetRouteWaypointCount()-1 までの間の整数値です。

戻り値

成功すれば表示オプション、失敗すれば-1 が返ります。

解説

表示オプションは、GPS の画面上のウェイポイントを表示するときに、シンボル、名称などを表示するかどうかを指定します。この値は GPS モデルによって同じ値で意味が異なります。

参照

UploadRoute メソッド

GarComm.ocx の内部バッファに保存されている全ルートデータを GPS にアップロードします。

BOOL UploadRoute(void);

パラメータ

無し

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

参照

UploadSingleRoute メソッド

GarComm.ocx の内部バッファに保存されているルートデータの 1 つを GPS にアップロードします。

```
BOOL UploadSingleRoute(  
    short rno  
);
```

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

参照

AddRoute メソッド

GarComm.ocx の内部バッファにルートを追加します。

```
BOOL AddRoute(  
    BSTR* name  
);
```

パラメータ

name

ルート名称を指定します。

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

参照

AddRouteWaypoint メソッド

ルートにウェイポイントを追加します。

BOOL AddRouteWaypoint(

long rno,

BSTR* name,

BSTR* comment,

double lat,

double lon,

double alt,

short smbl,

short dspl,

double dst

);

パラメータ

rno

ルート番号。0 ~ GetRouteCount()-1 までの間の整数値です。

name

ウェイポイント名称を指定します。

comment

ウェイポイントのコメントを指定します。

lat

緯度を指定します。

lon

経度を指定します。

alt

高度を指定します。

smbl

シンボル (アイコン) を指定します。

dspl

表示オプションを指定します。

dst

dst パラメータを指定します。

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

ウェイポイントはルート最後に追加されます。順番の変更などは上位で行ってください。

AddRouteWaypoint は同じ名称で既にウェイポイントがあるかどうかの確認は行いません。

参照

DeleteRouteAll メソッド

GarComm.ocx のルートデータ用の内部バッファを開放します。

```
void DeleteRouteAll(void);
```

パラメータ

無し

戻り値

無し

解説

このメソッドを実行すると、GarComm.ocx 内部の全ルートデータは消去されます。

参照

GetDatumCount メソッド

定義されている測地系の個数を取得します。

```
long GetDatumCount(void);
```

パラメータ

無し

戻り値

定義されている測地系の個数が返ります。

解説

定義されている測地系の一覧を取得するためには、GetDatumCount で個数を取得し、GetEntryDatumName で測地系の名称を取得してください。

参照

[GetEntryDatumName](#)

GetEntryDatumName メソッド

測地系の名称を指定します。

```
BSTR GetEntryDatumName(  
    long no  
);
```

パラメータ

no

測地系番号。0 ~ GetDatumCount()-1 の間の整数値です。

戻り値

測地系の名称が返ります。無効な値を指定したときには NULL が返ります。

解説

参照

GetEntryDatumName2 メソッド

東京測地系ならば、“ Bessel_1841 ” に相当する名称を取得します。

```
BSTR GetEntryDatumName2(  
    long no  
);
```

パラメータ

no

測地系番号。0 ~ GetDatumCount()-1 の間の整数値です。

戻り値

測地系の名称が返ります。無効な値を指定したときには NULL が返ります。

解説

参照

SelectDatum メソッド

測地系を指定します。

```
void SelectDatum(  
    long no  
);
```

パラメータ

no

測地系番号。0 ~ GetDatumCount()-1 の間の整数値です。

戻り値

無し

解説

測地系の変換は、GarComm.ocx の内部バッファに対する読み書きのときに行われます。
GarComm.ocx 内部では WGS84 測地系で保存されています。

参照

SelectDatumByName メソッド

測地系を名称で指定します。

```
long SelectDatumByName(  
    BSTR* MapDatumName  
);
```

パラメータ

MapDatumName

測地系の名称。GetEntryDatumName で取得した名称を指定します。

戻り値

選択された測地系の番号が返ります。測地系の番号は SelectDatum で同じ測地系を選択するために使用する番号になります。

解説

SelectDatumByName と GetEntryDatumName は逆変換の関係にあります。

参照

[SelectDatum](#)、[GetEntryDatumName](#)

GetDatumName メソッド

現在選択されている測地系の名称を取得します。

BSTR GetDatumName(void);

パラメータ

無し

戻り値

成功すると現在選択されている測地系の名称が返ります。失敗すると“ ERROR ”という文字列が返ります。

解説

参照

GetSelectedDatumNumber メソッド

現在選択されている測地系番号を取得します。

```
long GetSelectedDatumNumber(void);
```

パラメータ

無し

戻り値

成功すれば測地系番号、失敗すれば-1 が返ります。

解説

参照

GetMapDatumDeltaA メソッド

現在選択されている測地系の A パラメータを取得します。

```
double GetMapDatumDeltaA(void);
```

パラメータ

無し

戻り値

成功すれば現在選択されている測地系の A パラメータ、失敗すれば 0 が返ります。

解説

参照

GetMapDatumDeltaF メソッド

現在選択されている測地系の F パラメータを取得します。

```
double GetMapDatumDeltaF(void);
```

パラメータ

無し

戻り値

成功すれば現在選択されている測地系の F パラメータ、失敗すれば 0 が返ります。

解説

参照

GetMapDatumDeltaX メソッド

現在選択されている測地系の X パラメータを取得します。

```
double GetMapDatumDeltaX(void);
```

パラメータ

無し

戻り値

成功すれば現在選択されている測地系の X パラメータ、失敗すれば 0 が返ります。

解説

参照

GetMapDatumDeltaY メソッド

現在選択されている測地系の Y パラメータを取得します。

```
double GetMapDatumDeltaY(void);
```

パラメータ

無し

戻り値

成功すれば現在選択されている測地系の Y パラメータ、失敗すれば 0 が返ります。

解説

参照

GetMapDatumDeltaZ メソッド

現在選択されている測地系の Z パラメータを取得します。

```
double GetMapDatumDeltaZ(void);
```

パラメータ

無し

戻り値

成功すれば現在選択されている測地系の Z パラメータ、失敗すれば 0 が返ります。

解説

参照

TransformDatumWGS84IntoLocal メソッド

WGS84 から現在選択されている測地系への変換を行います。

```
BOOL TransformDatumWGS84IntoLocal(  
    double inLat,  
    double inLon,  
    double inAlt,  
    double* outLat,  
    double* outLon,  
    double* outAlt  
);
```

パラメータ

inLat

変換元の緯度を指定します。

inLon

変換元の経度を指定します。

inAlt

変換元の高度を指定します。

outLat

変換後の緯度が入る変数のポインタを指定します。

outLon

変換後の経度が入る変数のポインタを指定します。

outAlt

変換後の高度が入る変数のポインタを指定します。

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

GarComm.ocx の内部バッファの読み書き時に測地系の変換が行われますので、このメソッドは必ずしも使用する機会があるとは限りません。自分で測地系の変換が必要なときに使用します。

参照

TransformDatumLocalIntoWGS84 メソッド

現在の選択されている測地系から WGS84 に変換します。

BOOL TransformDatumLocalIntoWGS84(

```
double inLat,  
double inLon,  
double inAlt,  
double* outLat,  
double* outLon,  
double* outAlt  
);
```

パラメータ

inLat

変換元の緯度を指定します。

inLon

変換元の経度を指定します。

inAlt

変換元の高度を指定します。

outLat

変換後の緯度が入る変数のポインタを指定します。

outLon

変換後の経度が入る変数のポインタを指定します。

outAlt

変換後の高度が入る変数のポインタを指定します。

戻り値

成功すれば TRUE、失敗すれば FALSE が返ります。

解説

GarComm.ocx の内部バッファの読み書き時に測地系の変換が行われますので、このメソッドは必ずしも使用する機会があるとは限りません。自分で測地系の変換が必要なときに使用します。

参照

CalcDistance メソッド

2点間の距離を計算します。

```
double CalcDistance(  
    double lat1,  
    double lon1,  
    double lat2,  
    double lon2  
);
```

パラメータ

lat1

地点 1 の緯度を指定します。

lon1

地点 1 の経度を指定します。

lat2

地点 2 の緯度を指定します。

lon2

地点 2 の経度を指定します。

戻り値

地点 1 と地点 2 の距離がメートル[m]単位で返ります。

解説

参照

付録 1 測地系名称一覧

- | | | |
|--------------------------|-------------------------|--------------------------|
| (0) Adindan | (34) European 1979 | (68) NAD27 Bahamas |
| (1) Afgooye | (35) Finland Hayford | (69) NAD27 Canada |
| (2) AIN EL ABD 1970 | (36) G. Segara | (70) NAD27 Canal Zone |
| (3) Anna 1 Astro 1965 | (37) Gandajika Base | (71) NAD27 Caribbean |
| (4) ARC 1950 | (38) Geodetic Datum '49 | (72) NAD27 Central |
| (5) ARC 1960 | (39) GGRS 87 | (73) NAD27 CONUS |
| (6) Ascension Island '58 | (40) Guam 1963 | (74) NAD27 Cuba |
| (7) Astro B4 Sorol Atoll | (41) Gux 1 Astro | (75) NAD27 Greenland |
| (8) Astro Beacon ¥E¥ | (42) Herat North | (76) NAD27 Mexico |
| (9) Astro Dos 71/4 | (43) Hjorsey 1955 | (77) NAD27 San Salvadr |
| (10) Astronomic Stn '52 | (44) Hong Kong 1963 | (78) NAD83 |
| (11) Australian Geod '66 | (45) Hu-Tzu-Shan | (79) Nahrwn Masirah Is |
| (12) Australian Geod '84 | (46) Indian Bangladesh | (80) Nahrwn Saudi Arbia |
| (13) Austria NS | (47) Indian Mean Value | (81) Nahrwn United Arab |
| (14) Belgium 1950 | (48) Indian Thailand | (82) Naparima BWI |
| (15) Bellevue (IGN) | (49) Indonesia 74 | (83) Netherland Tri 21 |
| (16) Bermuda 1957 | (50) Ireland 1965 | (84) Nou Triag France |
| (17) Bogota Obsrvatry | (51) ISTS 073 Astro '69 | (85) Nou Triag Luxemb |
| (18) Bukit Rimpah | (52) Johnston Island | (86) Observatorio 1966 |
| (19) Camp Area Astro | (53) Kandawala | (87) Old Egyptian |
| (20) Campo Inchauspe | (54) Kerguelen Island | (88) Old Hawaii Kauia |
| (21) Canton Astro 1966 | (55) Kertau 1948 | (89) Old Hawaii Maui |
| (22) Cape | (56) L.C. 5 Astro | (90) Old Hawaii Mean |
| (23) Cape Canaveral | (57) Liberia 1964 | (91) Old Hawaii Oahu |
| (24) Carthage | (58) Luzon Mean Value | (92) Oman |
| (25) CH-1903 | (59) Luzon Mindanao | (93) Ord Srvy Grt Britn |
| (26) Chatham 1971 | (60) Luzon Philippines | (94) Pico De Las Nieves |
| (27) Chua Astro | (61) Mahe 1971 | (95) Pitcairn Astro 1967 |
| (28) Corrego Alegre | (62) Marco Astro | (96) Portuguese 1973 |
| (29) Danish Gi 1934 | (63) Massawa | (97) Potsdam |
| (30) Djakarta (Batavia) | (64) Merchich | (98) Prov So Amricn '56 |
| (31) Dos 1968 | (65) Midway Astro 1961 | (99) Prov So Chilean '63 |
| (32) Easter Island 1967 | (66) Minna | (100) Puerto Rico |
| (33) European 1950 | (67) NAD27 Alaska | (101) Qatar National |

(102) Qornoq	(110) South American '69	(118) Viti Levu 1916
(103) Reunion	(111) South Asia	(119) Wake-Eniwetok '60
(104) Rome 1940	(112) Southeast Base	(120) WGS 72
(105) RT 90	(113) Southwest Base	(121) WGS 84
(106) Santo (Dos)	(114) Tananarive Obsv	(122) Yacare
(107) Sao Braz	(115) Timbalai 1948	(123) Zanderij
(108) Sapper Hill 1943	(116) Tokyo	
(109) Schwarzeck	(117) Tristan Astro 1968	

Visual Basic サンプルプログラム

：トラックのダウンロード

```
Private Sub DownloadTrack_Click()  
    Dim lat As Double ' 緯度  
    Dim lon As Double ' 経度  
    Dim alt As Double ' 高度  
    Dim trk As Long  
    Dim leg As Long  
    Dim a As Long  
    Dim b As Long  
    Dim str As String  
    Dim id As Long  
    Dim ver As LogEventTypeConstants  
  
    ' GPS からモデル ID を呼び出す  
    GarComm1.ReadProductID id, ver  
    If -1 = id Then ' Read error  
        Debug.Print "Communication Error"  
        Exit Sub  
    End If  
  
    ' 読み出したモデル ID をコントロールにセットする  
    ' サポートしていない ID をセットするとエラー (0) になる  
    ' ここでデータ形式が同じ ID を強制的にセットすれば、サポートしていないモデルでも使用できる  
    If 0 = GarComm1.SetProductID(id) Then ' サポートしていないモデル ID  
        Exit Sub  
    End If  
  
    ' 測地系の選択  
    str = "Tokyo"  
    a = GarComm1.SelectDatumByName(str)  
  
    ' Track のダウンロード  
    GarComm1.DownloadTrack  
  
    Debug.Print "GetTrackCount"; GarComm1.GetTrackCount  
    For trk = 0 To GarComm1.GetTrackCount - 1  
        str = GarComm1.GetTrackName(trk)  
        Debug.Print str  
        Debug.Print "GetTrackSegmentCount"; GarComm1.GetTrackSegmentCount(trk)  
        For leg = 0 To GarComm1.GetTrackSegmentCount(trk) - 1  
            GarComm1.GetTrackSegmentCount (trk)  
            Debug.Print "GetTrackDataCount"; GarComm1.GetTrackDataCount(trk)  
            For k = 0 To GarComm1.GetTrackSegmentDataCount(trk, leg) - 1  
                GarComm1.GetTrackPosition trk, leg, k, lat, lon, alt  
                Debug.Print trk, leg, k, lat, lon, alt  
            Next k  
        Next leg  
    Next trk  
End Sub
```

：ウェイポイントのダウンロード

```
Private Sub DownloadWaypoint_Click()  
    Dim lat As Double ' 緯度  
    Dim lon As Double ' 経度  
    Dim alt As Double ' 高度  
    Dim wpt As Long  
    Dim id As Long
```

```

Dim ver As LogEventTypeConstants
Dim str As String

' GPS からモデル ID を呼び出す
GarComm1.ReadProductID id, ver
If -1 = id Then ' Read error
    Debug.Print "Communication Error"
    Exit Sub
End If

' 読み出したモデル ID をコントロールにセットする
' サポートしていない ID をセットするとエラー (0) になる
' ここでデータ形式が同じ ID を強制的にセットすれば、サポートしていないモデルでも使用できる
If 0 = GarComm1.SetProductID(id) Then ' サポートしていないモデル ID
    Exit Sub
End If

' 測地系の選択
GarComm1.SelectDatum (126)

' Waypoint のダウンロード
GarComm1.DownloadWaypoint

For wpt = 0 To GarComm1.GetWaypointCount - 1
    str = GarComm1.GetWaypointID(wpt)
    Debug.Print str,
        lat = GarComm1.GetWaypointLatitude(wpt)
        lon = GarComm1.GetWaypointLongitude(wpt)
        alt = GarComm1.GetWaypointAltitude(wpt)
    Debug.Print lat, lon, alt
Next wpt
End Sub

' ウェイポイントのアップロード
,
Private Sub UploadWaypoint_Click()
    Dim id As String
    Dim comment As String
    Dim lat As Double
    Dim lon As Double
    Dim alt As Double
    Dim smbl As Long
    Dim dspl As Long
    Dim dst As Double

    comment = "COMMENT"
    lat = 36
    lon = 141
    alt = 0
    smbl = 18
    dspl = 3
    dst = 0
    GarComm1.DeleteWaypointAll
    id = "0001"
    Call GarComm1.AddWaypoint(id, comment, lat, lon, alt, smbl, dspl, dst)
    id = "0002"
    Call GarComm1.AddWaypoint(id, comment, lat, lon, alt, smbl, dspl, dst)
    GarComm1.UploadWaypoint
End Sub

```