

次世代ロボット知能化技術開発プロジェクト
環境データコンポーネント
外部仕様書

2011年 2月 (Ver. A)

株式会社国際電気通信基礎技術研究所
知能ロボティクス研究所

環境データコンポーネント外部仕様書

・改訂履歴

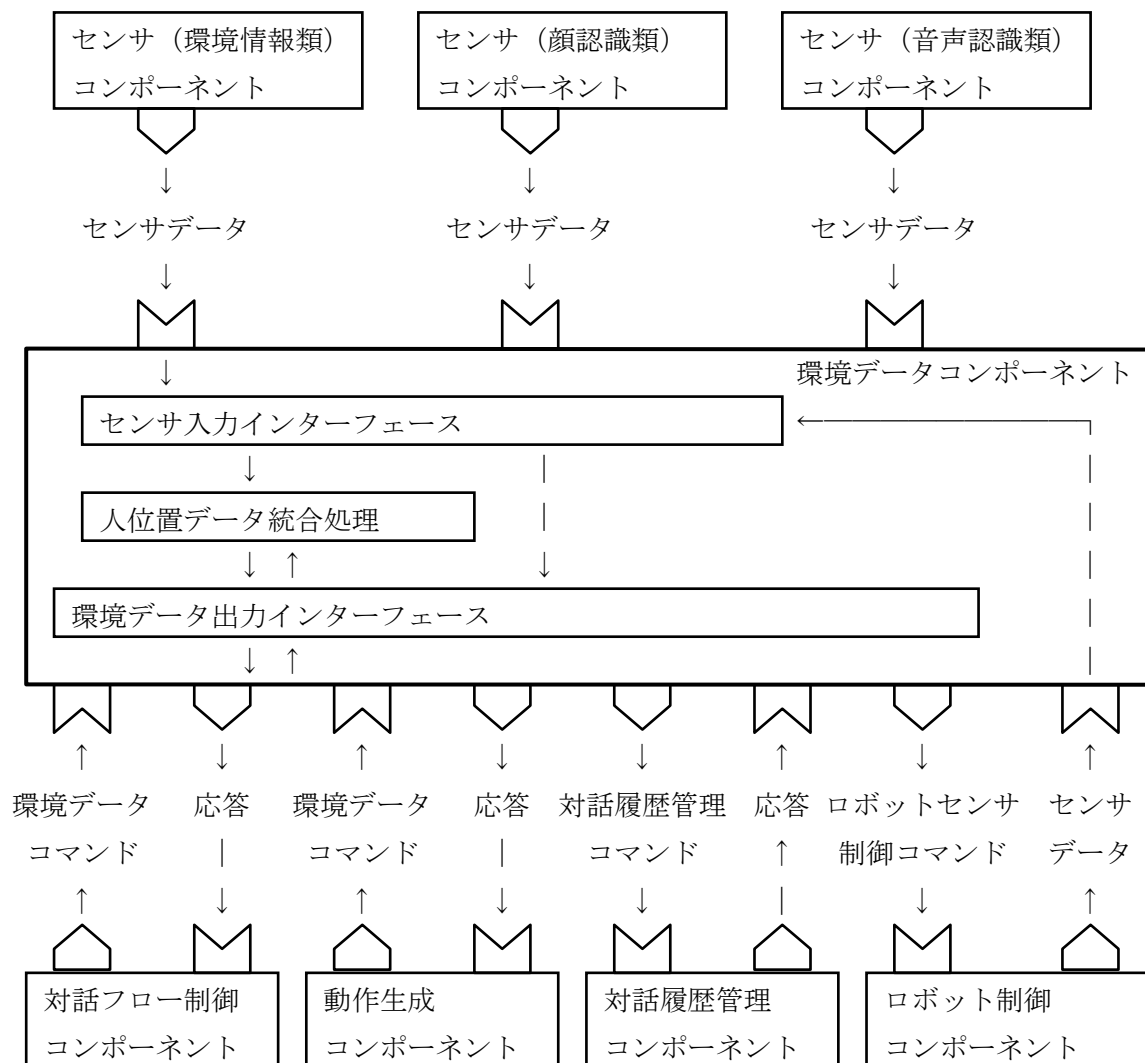
Ver.	改訂日付	改訂内容	作成
NC	2011-01-17	初版	岩崎
A	2011-02-21	誤記訂正	岩崎

目次

1. 環境データコンポーネントの目的	4
2. プログラム仕様.....	5
2.1 動作環境.....	5
2.2 データファイル.....	5
2.3 プログラムの使用方法.....	5
3. 外部インターフェース.....	5
4. 環境データコマンド形式.....	7
4.1 REQ コマンド.....	8
4.1.1 REQ コマンドで実装しているダク	9
4.2 SR コマンド.....	12
4.2.1 SR:BGN コマンド.....	12
4.2.2 SR:CAN コマンド.....	13
4.2.3 SR:RSP コマンド.....	13
4.2.4 SR:BHR コマンド.....	13
4.3 PS コマンド	14
4.3.1 PS:BS コマンド	14
4.4 TG コマンド	14
4.4.1 TG:MDS コマンド	14
4.4.2 TG:RLS コマンド	15
4.5 MG コマンド	15
4.5.1 MG:EC コマンド	15
5. その他コマンド形式	15
6. 環境情報データ設定ファイル(EnvData.conf)の詳細（参考）	15
6.1 設定例	16

1. 環境データコンポーネントの目的

環境データコンポーネントは、各種センサコンポーネントよりセンサデータを取り込んで統合し、その結果を上位層に出力する。



(1) センサ入力インターフェース

各種センサから入力されるセンサデータを処理し、内部で統合して利用可能な状態にする。

(2) 人位置データ統合処理

各種センサから入力される人位置に関するデータを統合し、ターゲットとなる人間を選択する。

(3) 環境データ出力インターフェース

上位層から送られる環境データコマンドを受信して解釈し、その応答を上位層に応答出力する。

2. プログラム仕様

2.1 動作環境

- ・使用 OS : WindowsXP
- ・ Visual C++ 2008 Express Edition
- ・ RT ミドルウェア (OpenRTM-aist-1.0.0)

2.2 データファイル

- ・ rtc.conf - RTM 設定ファイル。Initialize 時に読み込まれる。
- ・ EnvData.conf - 環境データ設定ファイル。
物体や場所の情報を登録する
Activate 時に読み込まれる。
漢字コード : UTF-8N
- ・ (mapdata.conf) - マップデータ設定ファイル。
Guide ビヘイビアで利用する場所を登録する。
Activate 時に読み込まれる。
漢字コード : UTF-8N

2.3 プログラムの使用方法

- (1) EnvDataModule のあるディレクトリで、Windows のコマンドプロンプトを開く。
※コマンドプロンプトの推奨プロパティ設定
現在のコードページ : 日本語 Shift-JIS
フォント : MS ゴシック
- (2) UTF-8N 漢字コードでモニタする場合、コマンドプロンプトにて、"chcp 65001"と入力する。
- (3) " EnvDataModuleComp"で起動する。
- (4) RTSystemEditor で接続し、Activate して使用する。
- (5) RTSystemEditor で Dectivate して Exit した後、Ctrl-C で終了する。

3. 外部インターフェース

(1)環境データコマンド入力ポート 1

ポート名 : ED_EDC1in
型 : InPort TimedString×1
接続先 : 対話フロー制御コンポーネント
漢字コード : UTF-8N

(2)環境データコマンド応答出力ポート 1

ポート名 : ED_EDCR1out
型 : OutPort TimedString×1
接続先 : 対話フロー制御コンポーネント

環境データコンポーネント外部仕様書

漢字コード：UTF-8N

(3)環境データコマンド入力ポート 2

ポート名：ED_EDC2in

型：InPort TimedString×1

接続先：動作生成コンポーネント MG_EDCout

漢字コード：UTF-8N

(4)環境データコマンド応答出力ポート 2

ポート名：ED_EDCR2out

型：OutPort TimedString×1

接続先：動作生成コンポーネント MG_EDCRin

漢字コード：UTF-8N

(5)対話履歴管理コマンド出力ポート

ポート名：ED_HMCout

型：OutPort TimedString×1

接続先：対話履歴管理コンポーネント HM_HMC2in

漢字コード：UTF-8N

(6)対話履歴管理コマンド応答入力ポート

ポート名：ED_HMCRin

型：InPort TimedString×1

接続先：対話履歴管理コンポーネント HM_HMCR2out

漢字コード：UTF-8N

(7)ロボット制御センサコマンド出力ポート

ポート名：ED_RSCout

型：OutPort TimedString×1

接続先：ロボット制御コンポーネント RM_RSCin

(8)ロボット制御センサデータ入力ポート

ポート名：ED_RSDin

型：InPort TimedLongSeq[n]×1

接続先：ロボット制御コンポーネント RM_RSDout

(9)音声認識コマンド出力ポート

ポート名：ED_SRCout

型：OutPort TimedString×1

環境データコンポーネント外部仕様書

接続先：音声認識コンポーネント

(10)音声認識コマンド応答入力ポート

ポート名：ED_SRCRin

型：InPort TimedString×1

接続先：音声認識コンポーネント

漢字コード：UTF-8N

(11)顔認識結果入力ポート

ポート名：ED_FRRin

型：InPort TimedString×1

接続先：顔認識コンポーネント

漢字コード：UTF-8N

(12)個人同定結果入力ポート（対外）

ポート名：ED_FDRin

型：InPort TimedString×1

接続先：個人同定コンポーネント

漢字コード：UTF-8N

(13)環境状況認識結果入力ポート

ポート名：ED_HTBRin

型：InPort TimedString×1

接続先：環境状況認識コンポーネント

(14)人物追跡結果入力ポート

ポート名：ED_HTTTrin

型：InPort TimedString×1

接続先：人物追跡コンポーネント

(15)インタラクション・リモート・コントローラ専用 TCP/IP ソケット

接続先：インタラクション・リモート・コントローラ接続用ポート

接続先ポート番号：5573

4. 環境データコマンド形式

環境データコマンド入力ポート（対話フロー制御コンポーネントと動作生成コンポーネントに接続）から入力されるコマンドと、その応答について記述する。

環境データコマンドは、次の種類がある。

環境データコンポーネント外部仕様書

- ・REQ コマンド：指定されたタグの情報を取得(BG,MG->ED)
- ・SR コマンド：音声認識に関するコマンド／情報を取得(BG,MG->ED)
- ・PS コマンド：PrimitiveServer に関するコマンド／情報を取得(BG,MG->ED)
- ・TG コマンド：ターゲット選択に関するコマンド(BG->ED)
- ・MG コマンド：MotionGenerator 状態通知に関するコマンド(MG->ED)

4.1 REQ コマンド

- ・機能：指定されたタグの情報をテキスト形式で戻す。
- ・入力コマンド形式：<REQ tag,param1,param2,...,paramN>
- ・入力パラメータ：tag - タグ名（31 文字以下の英数字とアンダーバー）
- ・応答出力：以下のいずれか

"<REQ tag,type,n,data1,data2,...,dataN>" - 正常終了

tag - タグ名（31 文字以下の英数字とアンダーバー）

タグ名の先頭に#が付加されているものは、固定値データを表す。

type - データタイプ（int32 型整数）

今回の実装例では、以下の通り。

0 = データなし

1 = 人データ

2 = Robot データ

3 = 物体データ

4 = 場所データ

5 = 数値データ

6 = 文字データ

valid - valid フラグ

0 = invalid (データ無効)

1 = valid (データ有効)

n - データ(data)の個数

data - データ（int32 型整数、もしくは文字列値）

文字列値は、""で囲む。

今回の実装例では、以下の通り。

座標値は、int32 型整数で表現できるもの。

座標系は、絶対座標系である。

"<REQ ERROR>" - 異常終了

"<ERROR>" - 原因不明のエラー

- ・type=0（データなし）の場合の例
 - ・入力例："<REQ dare>"
 - ・出力例："<REQ dare,0,0,0>"
- ・type=1（人データ）の場合の例
 - ・入力例："<REQ Human>"

環境データコンポーネント外部仕様書

- ・ 出力例 : "<REQ Human,1,1,5,1,2000,0,0,0>"
- ・ type=2 (Robot データ) の場合の例
 - ・ 入力例 : "<REQ Robot>"
 - ・ 出力例 : "<REQ Robot,2,1,5,1,2000,0,0,0>"
- ・ type=3 (物体データ) の場合の例
 - ・ 入力例 : "<REQ Object1>"
 - ・ 出力例 : "<REQ #Object1,3,1,4,2000,0,0,0>"
- ・ type=4 (場所データ) の場合の例
 - ・ 入力例 : "<REQ Place1>"
 - ・ 出力例 : "<REQ #Place1,4,1,3,2000,0,0,0>"
- ・ type=5 (数値データ) の場合の例
 - ・ 入力例 : "<REQ num1>"
 - ・ 出力例 : "<REQ #num1,5,1,1,100>"
- ・ type=6 (数値データ) の場合の例
 - ・ 入力例 : "<REQ text1>"
 - ・ 出力例 : "<REQ #text1,6,1,1,\"test\">"

4.1.1 REQ コマンドで実装しているダク

(1)Human

内容 : 人の位置

応答出力フォーマット :

tag = "Human" - タグ名
type = 1 - データタイプ(人データ)
valid = 1 - valid フラグ(1=valid)
n = 5 - データ(data)の個数
id, - 人 ID
data1,2,3 - 体の中心座標 X,Y,Z [mm:int32]
data4 - 体の前方方向の向き [deg:int]

(2)HumanIdXXXXXXXX (XXXXXXXX は人 ID)

内容 : 人 ID で指定された人の位置

応答出力フォーマット :

tag = "HumanIdXXXXXXXX" - タグ名
type = 1 - データタイプ(人データ)
valid = 1 - valid フラグ(1=valid)
n = 5 - データ(data)の個数
id, - 人 ID
data1,2,3 - 体の中心座標 X,Y,Z [mm:int32]
data4 - 体の前方方向の向き [deg:int]

(3)HumanCount

内容：検出された人の数

応答出力フォーマット：

tag = "HumanCount" - タグ名
type = 5 - データタイプ(数値データ)
valid = 1 - valid フラグ(1=valid)
n = 1 - データ(data)の個数
data1 - 人検出数 [int32]

(4)Robot

内容：ロボットの位置

応答出力フォーマット：

tag = "Robot" - タグ名
type = 2 - データタイプ(Robot データ)
valid = 1 - valid フラグ(1=valid)
n = 5 - データ(data)の個数
id, - ロボット ID
data1,2,3 - 体の中心座標 X,Y,Z [mm:int32]
data4 - 体の前方方向の向き [deg:int]

(5)RobotStatus

内容：ロボットの状態

応答出力フォーマット：

tag = "RobotStatus" - タグ名
type = 5 - データタイプ(数値データ)
valid = 1/0 - valid フラグ(1=valid)
n = 1～99 - データ(data)の個数
data1 - センサデータ 1 [int32]
:
dataN - センサデータ N [int32]

(6)FaceCount

内容：検出された顔の数

応答出力フォーマット：

tag = "FaceCount" - タグ名
type = 5 - データタイプ(数値データ)
valid = 1 - valid フラグ(1=valid)
n = 1 - データ(data)の個数

環境データコンポーネント外部仕様書

data1 - 顔検出数 [int32]

(7)FaceData

内容：検出された顔のデータ

応答出力フォーマット：

tag = "FaceData" - タグ名
type = 5 - データタイプ(数値データ)
valid = 1/0 - valid フラグ(1=valid)
n = 8 - データ(data)の個数
data1 - 顔の左上座標 X[int32]
data2 - 顔の左上座標 Y[int32]
data3 - 顔の右上座標 X[int32]
data4 - 顔の右上座標 Y[int32]
data5 - 顔の左下座標 X[int32]
data6 - 顔の左下座標 Y[int32]
data7 - 顔の右下座標 X[int32]
data8 - 顔の右下座標 Y[int32]

(8)FaceUid

内容：検出された顔の個人識別 ID

応答出力フォーマット：

tag = "FaceId" - タグ名
type = 6 - データタイプ(文字データ)
valid = 1/0 - valid フラグ(1=valid)
n = 1 - データ(data)の個数
data1 - 顔の個人識別 ID [string]

(9)FaceId

内容：検出された顔の ID

応答出力フォーマット：

tag = "FaceId" - タグ名
type = 5 - データタイプ(数値データ)
valid = 1/0 - valid フラグ(1=valid)
n = 1 - データ(data)の個数
data1 - 顔の ID(個人識別 ID の先頭数値)[int32]

(10)HumanGaze

内容：視線動作検出された人の状態(MotionGenModule の implicit で使用)

応答出力フォーマット：

環境データコンポーネント外部仕様書

tag = "HumanGaze" - タグ名
type = 5 - データタイプ(数値データ)
valid = 1 - valid フラグ(1=valid)
n = 3 - データ(data)の個数
data1 - 視線動作検出の状態(1=視線動作検出中) [:int]
data2 - 視線動作検出の yaw 角度 [deg:int]
data3 - 視線動作検出の pitch 角度 [deg:int]

(11)EnvData.conf に定義されている物体タグ

内容：物体の位置

応答出力フォーマット：

tag = 物体名 - タグ名
type = 3 - データタイプ(物体データ)
valid = 1 - valid フラグ(1=valid)
n = 4 - データ(data)の個数
data1,2,3 - 物体の中心座標 X,Y,Z [mm:int32]
data4 - 物体の前方方向の向き [deg:int]

(12)EnvData.conf に定義されている場所タグ

内容：場所の位置

応答出力フォーマット：

tag = 場所名 - タグ名
type = 4 - データタイプ(場所データ)
valid = 1 - valid フラグ(1=valid)
n = 3 - データ(data)の個数
data1,2,3 - 場所の中心座標 X,Y,Z [mm:int32]

4.2 SR コマンド

音声認識に関するコマンドである。

音声認識モジュールが接続されている時に有効である。

4.2.1 SR:BGN コマンド

- 機能：指定された時間まで音声認識を行い、マッチング条件に合うものが認識された時点で音声認識を終了する。

音声認識結果は、SR:RSP コマンドにより取得すること。

- コマンド入力形式：

<SR:BGN rtime>

<SR:BGN rtime,mtext>

- 入力パラメータ：

環境データコンポーネント外部仕様書

rtime - 最大音声取込時間[s]。(1～99)

mtext - マッチング文字列。

"/"で文節を区切って、OR 条件で列挙できる。

マッチング文字列の指定のない場合は、**rtime** で指定された時間まで取り込みを行う。

・入力例：

<SR:BGN 5>

<SR:BGN 5,/東京/京都/大阪/>

・応答出力：以下のいずれか

<SR:BGN OK> - 正常終了(動作終了ではなく、BGN コマンド受信完了)

<SR:BGN ERROR> - 異常終了

<ERROR> - 原因不明のエラー

4.2.2 SR:CAN コマンド

・機能：音声認識を停止する。

・コマンド入力形式：<SR:CAN>

・入力パラメータ：なし

・入力例：<SR:CAN>

・応答：なし

4.2.3 SR:RSP コマンド

・機能：音声認識結果を取得する。

・コマンド入力形式：<SR:RSP>

・入力パラメータ：なし

・入力例：<SR:RSP>

・応答出力：以下のいずれか

<SR:RSP NO n> - 音声認識実行中のため結果なし

n=0: 人は現在発話していない

n=1: 人は現在発話中

<SR:RSP OK 結果文字列> - 正常終了

<SR:RSP ERROR> - 異常終了

<ERROR> - 原因不明のエラー

結果文字列は、以下のように文節毎に"/"で区切られる。

<SR:RSP OK /何処/から来た/の/>

<SR:RSP OK /京都/だよ//いや/東京/だよ/>

<SR:RSP OK > … うまく認識できないと結果文字列がない場合もある。

4.2.4 SR:BHR コマンド

・機能：音声認識を明示的に実行（SR:BGN や SR:SEL）していない時に、バックグラウンド

環境データコンポーネント外部仕様書

で実行している背景音声認識結果を取得する。

一度読み出すと、背景音声認識結果はクリアされることに注意。

- ・ コマンド入力形式：<SR:BHR>

- ・ 入力パラメータ：なし

- ・ 入力例：<SR:BHR>

- ・ 応答出力：以下のいずれか

<SR:BHR OKn 結果文字列> - 正常終了

n=0: 人は現在発話していない

n=1: 人は現在発話中

<SR:BHR ERROR> - 異常終了

<ERROR> - 原因不明のエラー

結果文字列は、以下のように文節毎に"/"で区切られる。

<SR:BHR OK0 /何処/から来た/の/>

<SR:BHR OK1 /京都/だよ/>

<SR:BHR OK1 > … 結果文字列がない場合もある。

4.3 PS コマンド

PrimitiveServer に関するコマンドである。

PrimitiveServer モジュールが接続されている時に有効である。

4.3.1 PS:BS コマンド

- ・ 機能：基本 PrimitiveServer の情報を取得する。

- ・ コマンド入力形式：<PS:BS category>

- ・ 入力パラメータ：category - カテゴリ文字列。

- ・ 入力例：<PS:BS Category>

- ・ 応答出力：以下のいずれか

<PS:BS OK 結果文字列> - 正常終了

<PS:BS OK > - 正常終了(結果文字列の無い場合)

<PS:BS ERROR> - 異常終了

<ERROR> - 原因不明のエラー

4.4 TG コマンド

ターゲット選択に関するコマンドである。

4.4.1 TG:MDS コマンド

- ・ 機能：ターゲット選択モードを指定する。

- ・ コマンド入力形式：<TG:MDS mode,param>

- ・ 入力パラメータ：mode - ターゲット選択モード。(0~9)

0 - なし

環境データコンポーネント外部仕様書

1 - 一人用顔認識

2 - UNR 用

parm - パラメータ

- ・ 入力例 : <TG:MDS 1,0>
- ・ 応答出力 : 以下のいずれか
 - "<TG:MDS OK>" - 正常終了
 - "<TG:MDS ERROR>" - 異常終了
 - "<ERROR>" - 原因不明のエラー

4.4.2 TG:RLS コマンド

- ・ 機能 : ターゲットされている状態を解除する。
- ・ コマンド入力形式 : <TG:RLS>
- ・ 入力パラメータ : なし
- ・ 入力例 : <TG:RLS>
- ・ 応答出力 : 以下のいずれか
 - "<TG:RLS OK>" - 正常終了
 - "<TG:RLS ERROR>" - 異常終了
 - "<ERROR>" - 原因不明のエラー

4.5 MG コマンド

MotionGenerator 内部状態通知に関するコマンドである。

4.5.1 MG:EC コマンド

- ・ 機能 : MotionGenerator のアイコンタクト動作状態を ED 側に通知する。
- ・ コマンド入力形式 : <MG:EC state>
- ・ 入力パラメータ : state - アイコンタクトの状態(OFF/ON)
- ・ 入力例 : <MG:EC ON>
- ・ 応答出力 : なし

5. その他コマンド形式

以下のコマンドについては、それぞれの仕様書を参照のこと。

- ・ ロボットセンサ制御コマンド(ED->MG) … ロボット制御コンポーネント仕様書
- ・ 対話履歴管理コマンド(ED->HM) … 対話履歴管理コンポーネント仕様書

6. 環境情報データ設定ファイル(EnvData.conf)の詳細 (参考)

- ・ コメント行は、先頭に"/を書くか、空白行とする。
- ・ 行末コメント行は、コメントの前に"/を書く。
- ・ タグ名は、31 文字以下の英数字とアンダーバーである。

環境データコンポーネント外部仕様書

- ・数値は、int32 型整数値である。
- ・数値中の "," と " " と "¥n" は、数値の区切り記号と見なされる。
- ・文字列値は、""で囲む。
- ・文字列中には "" は使用不可。
- ・更新しない固定値データは、タグ名の前に#を付加する。
- ・位置座標は、基本的には絶対座標系(Xa,Ya,Za,Da)とする。

6.1 設定例

--- ここから ---

// サンプルファイル定義

```
<Human          // タグ名(#=固定値データ)
1,              // データタイプ(1=人, 2=Robot, 3=物体, 4=場所)
1,              // valid フラグ(0=invalid, 1=valid)
4,              // データ数
2000,0,0,0      // 位置 X,Y,Z,角度
>              // タグ終了
```

```
<Robot          // タグ名(#=固定値データ)
2,              // データタイプ(1=人, 2=Robot, 3=物体, 4=場所)
1,              // valid フラグ(0=invalid, 1=valid)
4,              // データ数
0,0,0,0         // 位置 X,Y,Z,角度
>              // タグ終了
```

```
<#Object        // タグ名(#=固定値データ)
3,              // データタイプ(1=人, 2=Robot, 3=物体, 4=場所)
1,              // valid フラグ(0=invalid, 1=valid)
4,              // データ数
1000,-1000,0,0  // 位置 X,Y,Z,角度
>              // タグ終了
```

```
<#Place         // タグ名(#=固定値データ)
4,              // データタイプ(1=人, 2=Robot, 3=物体, 4=場所)
1,              // valid フラグ(0=invalid, 1=valid)
3,              // データ数
1000,-500,0     // 位置 X,Y,Z
>              // タグ終了
```

--- ここまで ---