

# Titan / 制御用プログラム開発資料

設計：白井達也

1999 年 6 月 1 6 日

（広島大学）

# Table of contents

|          |                                 |           |
|----------|---------------------------------|-----------|
| <b>1</b> | <b>履歴.....</b>                  | <b>3</b>  |
| 1-1      | ORIGINAL VERSION .....          | 3         |
| 1-2      | VER.1.0 (TIP 対応準備版) .....       | 4         |
| 1-3      | VER.2.0 足裏センサの追加およびコマンド追加 ..... | 5         |
| 1-4      | VER.3.0 (VISUAL C++対応版).....    | 5         |
| <b>2</b> | <b>システムの基本モード.....</b>          | <b>6</b>  |
| 2-1      | 手動モード .....                     | 7         |
| 2-1-1    | 手動モードで可能な操作 .....               | 7         |
| 2-1-2    | 手動モードに入った時に行われる処理 .....         | 7         |
| 2-1-3    | 手動モード中に行われる処理 .....             | 7         |
| 2-2      | 原点復帰モード .....                   | 7         |
| 2-2-1    | 原点復帰モードで可能な操作 .....             | 7         |
| 2-2-2    | 原点復帰モードに入った時に行われる処理 .....       | 7         |
| 2-2-3    | 原点復帰モード中に行われる処理 .....           | 7         |
| 2-3      | 自動モード .....                     | 8         |
| 2-3-1    | 自動モードで可能な操作 .....               | 8         |
| 2-3-2    | 自動モードに入った時に行われる処理 .....         | 8         |
| 2-3-3    | 自動モード中に行われる処理 .....             | 8         |
| 2-4      | 非常停止モード .....                   | 9         |
| 2-4-1    | 非常停止モードで可能な操作 .....             | 9         |
| 2-4-2    | 非常停止モードに入った時に行われる処理 .....       | 9         |
| 2-4-3    | 非常停止モード中に行われる処理 .....           | 9         |
| <b>3</b> | <b>ロボット制御モード.....</b>           | <b>10</b> |
| <b>4</b> | <b>メニュー構成 .....</b>             | <b>11</b> |
| 4-1      | TOP 項目 .....                    | 11        |
| 4-2      | PARAMETER 細目.....               | 12        |
| <b>5</b> | <b>各種ファイルの書式.....</b>           | <b>13</b> |
| 5-1      | 原点記憶ファイル(TIT_ORG.INI) .....     | 13        |
| <b>6</b> | <b>デバッグ方法 .....</b>             | <b>13</b> |
| <b>7</b> | <b>将来的な展望 .....</b>             | <b>14</b> |

## 1 履歴

### 1 - 1 Original Version

**製作年月日** : 1998 年度

**設計 / 製作者** : 白井, 井上

以下 4 つのモードをもつ制御用プログラムのプロトタイプを製作 .

1. 手動モード : Leg\_controller による指定した脚の X,Y,Z 座標の操作
2. 原点復帰モード : 原点位置のクリア , 初期姿勢への移動
3. 自動モード : AUTO プログラム ( ネイティブコード ) による自動運転可能 .
  - 脚先接地 ( 静歩行 )
  - 胴体接地 ( 1 足ずつ , バタフライ : 4 脚同時 )
4. 非常停止モード : 非常停止履歴ログ生成

Off-Line ( 画面上 ) で Titan の動作を確認できる OFFLINE\_DEBUG バージョンの生成も可能 .

**製作年月日** : 1999 年 6 月

**設計 / 製作者** : 白井

- ・ 定数のファイル(TITAN.INI)からの読み込みを可能にする .
  - ・ A/D ボード上のタイマを用いたサンプリングタイムの均一化 , およびウォッチドグ機能 .
  - ・ 関節レベルの制御から脚先ベクトルによる制御へ拡張 . 同時に最大速度 , 最大加速度の制限を可能とする .
  - ・ 変数名 , 定数名の統一を図る .
  - ・ 現在の脚先座標のみの目標指令をトルク指令 , コンプライアンス指令 , 関節角度指令にも対応できるように , 制御モードの拡張を行う .
  - ・ ログ記録および出力機能の実現 .
  - ・ プログラム単位を現在の titan.c/h , tit\_disp.c/h , strategy.c/h に加え , tit\_hard.c/h に分割することで , ハードウェア依存する部分を独立させる . 同時に strategy.c/h を単純に差し替えるだけでコンパイルできるように独立性を高める .
  - ・ 原点位置 (ポテンショメータ) のファイルへの書き出しおよび読み込みを可能とする .
  - ・ バグの修正 ( SystemMessage 画面表示関連 , 原点復帰後の F4 のズレ=all\_stop()について . などなど )
  - ・ 各モード遷移(mode\_transfer)のドキュメント化を行い , 仕様を整理する .
  - ・ ロボット制御部(robot\_controller)のドキュメント化を行い , 仕様を整理する .
  - ・ system\_message の使用方法を高機能化 , 高性能化する . 具体的には , バッファを介して画面表示することで , 簡単に再描画を可能とする .
  - ・ OFFLINE\_DEBUG 時には , titan.angle を angle\_err に基づいて feedback していたのを order(モータへの出力 : 単位[V])に基づいて , sensor.digit へ feedback することで , よりハードウェアに近いエミュレーションを可能とする .
  - ・ メニュー ( 非常停止モードでのみ使用可能 ) 機能 . メニューにより ,
    - ☐ 各種定数の変更 ( ゲイン , サンプリングタイムなど )
    - ☐ 画面に表示する情報(textmode)の切り替え
    - ☐ 子プロセス ( command.com ) の起動
    - ☐ Leg\_controller の機能選択 ( 脚先座標 ← → 関節角度 )
- などなど .

### 1 - 3 Ver.2.0 足裏センサの追加およびコマンド追加

**製作年月日** : 2000 年 2 月 26 日

**設計 / 製作者** : 白井

足裏圧力センサや触角センサなど ,I / O の拡張に対応.それに合わせて ,TIP のバージョンを 2 . 0 に上げる. コマンドの追加・修正の詳細については , TIP 仕様書参照.

### 1 - 4 Ver.3.0 (Visual C++対応版)

**製作年月日** : 未定

**設計 / 製作者** : 未定

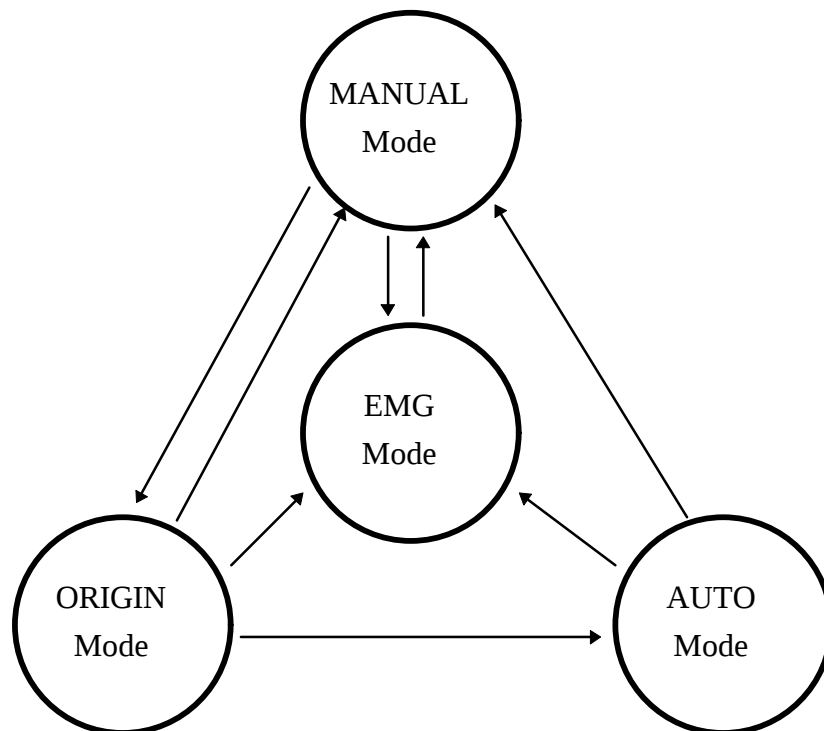
huge ポインタから開放される .

32bit 命令の使用によって,もしかしたらもっと速く動くようになるかも知れない . ただしハードウェア割り込み ( H D D など ) やマルチタスクの功罪 ( オーバーヘッドの増大 , あるいは不均一 ) によるトラブル発生の可能性は高い . 仮想記憶なし , で , どこまで動くか ?

## 2 システムの基本モード

本ロボット制御プログラムは、基本的に以下4つのモードを持ち、状態遷移が制限されている。

- (1) 手動モード               : 手動操作ボックスによる脚先（または関節）の操作が可能なモード。
- (2) 原点復帰モード       : Titan を初期姿勢に移動させるためのモード。
- (3) 自動モード             : Titan を AUTO/TIP プログラムにて自動運転させるためのモード
- (4) 非常停止モード       : 非常停止要因が発生した際に Titan を安全に強制停止させるモード



非常停止モードには、全ての状態から遷移する可能性がある。たとえば関節角度がリミットに達した場合や演算エラー、TIP プログラムのランタイムエラーなどが発生した場合である。非常停止モードから抜けるときは、必ず手動モードに戻る。自動モードへは、原点復帰完了記憶がONの時のみ（原点復帰モードを介して）入ることができる。また原点復帰モードへは手動モードからしか入れない。原点復帰モード、自動モードからは、いつでも手動モードへ入ることが可能である。

## 2-1 手動モード

### 2-1-1 手動モードで可能な操作

### 2-1-2 手動モードに入った時に行われる処理

- 原点復帰動作フラグをクリアする．
- 原点復帰完了記憶をクリアする．
- 自動運転動作中フラグをクリアする．
- 自動運転プログラム No.を未選択状態にクリアする．
- 全出力をゼロ[V]とする．
- PTP フラグをクリアする．
- 制御モードは脚先座標（あるいは関節角度：leg\_control のモードに従う）モードに遷移し,現在の位置（角度）を目標値として設定しなおす．
- ログの記録を一時停止する．

### 2-1-3 手動モード中に行われる処理

（作成中）

## 2-2 原点復帰モード

### 2-2-1 原点復帰モードで可能な操作

### 2-2-2 原点復帰モードに入った時に行われる処理

- 原点復帰動作中フラグをリセットする．
- 自動運転動作中フラグをリセットする．
- PTP フラグをリセットする．
- 脚先制御モードに切り替える．

### 2-2-3 原点復帰モード中に行われる処理

（作成中）

## 2-3 自動モード

### 2-3-1 自動モードで可能な操作

### 2-3-2 自動モードに入った時に行われる処理

- 自動運転中フラグをリセット
- PTP フラグをリセット
- 脚先制御モードを FOOT\_PTP へ変更 .
- 自動運転ステータスを AT\_INITIAL にリセット .
- 自動運転一時停止 / 停止要求フラグをリセット

### 2-3-3 自動モード中に行われる処理

( 作成中 )

## 2-4 非常停止モード

### 2-4-1 非常停止モードで可能な操作

### 2-4-2 非常停止モードに入った時に行われる処理

```
        sys.origin = FALSE;                /* 原点復帰未完とする          */
        all_0V();                          /* D A出力を 0[V] に, 目標 Reset    */
        origin_st.initial_running = FALSE; /* 原点復帰動作フラグをクリア      */
        auto_st.running = FALSE;          /* 念のために                      */
        robot.ptp_running[F1] = FALSE;    /* 念のために                      */
        robot.ptp_running[F2] = FALSE;    /* 念のために                      */
        robot.ptp_running[F3] = FALSE;    /* 念のために                      */
        robot.ptp_running[F4] = FALSE;    /* 念のために                      */
        LEG_CONTROL_MODE_F1(NO_CONTROL);
        LEG_CONTROL_MODE_F2(NO_CONTROL);
        LEG_CONTROL_MODE_F3(NO_CONTROL);
        LEG_CONTROL_MODE_F4(NO_CONTROL);
    }
    sys.mode = new_mode;
}
```

•

### 2-4-3 非常停止モード中に行われる処理

(作成中)

### 3 ロボット制御モード

Titan には、大きく分けて以下二つの制御モードがある。

#### (1) 脚先制御モード<sup>1</sup>

- 脚先座標制御モード
- 脚先速度制御モード

#### (2) 関節制御モード

- 関節角度制御モード
- 関節角速度制御モード
- トルク制御モード

脚先座標 / 速度制御モードは脚毎に設定可能である。それに対して関節角度 / 角速度制御モードとトルク制御モードは各関節毎に指定が可能である。ただし一つの脚に対して脚先制御モードと関節制御モードを混在させることはできない。<sup>2</sup>

関節制御モード状態の脚を脚先制御モードに切り替える場合には問題が発生しないが、脚先制御モード下にある脚に対して、そのうちの一つの関節を関節制御モードに移した場合には問題が発生する。指定されなかった関節はどうすれば良いのか？ 解決策の一つは、関節制御モードへの変更を一つの脚に対して行うときは、その脚の全ての関節に対して指定を行わなければならない、という手である。ただしこれはかなり使う側（特に TIP プログラムを利用する側）からすると、非常に面倒くさい。いちいち脚のモードを切り替えて、それからコマンド(ANGLE コマンドなど)を記述するのか。モード切り替えを忘れたら、そのたびに非常停止するのか？

そこで以下の暗黙のルールを設ける。脚先制御モードから関節制御モードに移行する際に、元のモードに応じて指定されなかった関節のモードと状態が状態遷移の立ち上がりパルストリガとして設定しなおされる。

| 脚先制御モード→ | 関節制御モード  | 残りの関節の制御モード | 目標設定  |
|----------|----------|-------------|-------|
| 座標制御モード  | 角度制御モード  | 角度制御モード     | 現在の角度 |
|          | 角速度制御モード | 角度制御モード     | 現在の角度 |
|          | トルク制御モード | 角度制御モード     | 現在の角度 |
| 速度制御モード  | 角度制御モード  | 角度制御モード     | 現在の角度 |
|          | 角速度制御モード | 角度制御モード     | 現在の角度 |
|          | トルク制御モード | 角度制御モード     | 現在の角度 |

<sup>1</sup> 将来的には脚先力制御、ハイブリッド制御、あるいは第 2 関節の力制御、ハイブリッド制御などが実装されることもあるかも知れないが、さすがにそこまでは先読みをしていない。

<sup>2</sup> これはごく当然なことである。

要するに，脚先制御モードから（関節のうちのどれか一つが）関節制御モードになったならば，それ以外の関節はすべて関節角度制御モードに移行し，目標値を現在の角度（つまり停止）することとする．

## 4 メニュー構成

メニューによる拡張機能は，非常停止モードにおいて行使可能である．

### 4-1 TOP 項目

|   |                |                                                                                                                                                                                |
|---|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| M | <MANUAL mode>  | 1 Leg Position Control mode<br>2 Joint Angle Control mode<br>3 Joint Torque Control mode(未対応)<br>4 Leg Velocity Control mode(未対応)<br>5 Joint Omega Control mode(未対応)<br>Q Quit |
| O | <ORIGIN mode>  | 1 Write org-file<br>2 Read org-file<br>3 Display Origin<br>Q Quit                                                                                                              |
| A | <AUTO mode>    | 1 登録されているAUTOプログラムの表示<br>2 Write Variable Table<br>3 Change TIP-Filename<br>Q Quit                                                                                             |
| E | <EMG mode>     | 1 EMG_ON/OFF<br>Q Quit                                                                                                                                                         |
| P | <Parameter>    | 1 Display<br>2 Change<br>3 Write ini-file<br>Q Quit                                                                                                                            |
| D | <Display Mode> | 1 Normal<br>2 Position Only<br>3 Angle Only<br>4 Position & DA<br>5 TIP/Variable Table                                                                                         |

表 4-1 メニュー構成（TOP 項目）

## 4-2 Parameter 細目

さらに ,Parameter の構成は ,Display,Change とともに 基本的に以下のような構成になっている .  
Write はあるが , Read が無い理由は,もし新しい設定ファイルを読み込みたいならば,再起動を掛けるべきであるとの判断に基づいている .

|   |             |                                     |                                                                                                                                               |                                                                                                                                                                                                                                       |
|---|-------------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| E | <Error>     | 1<br>2<br>3<br>Q                    | <Position Error><br><Angle Error><br><Torque Error><br>Quit                                                                                   | F1, F2, F3, F4, Quit<br>(JOINT) / Quit<br>(JOINT) / Quit                                                                                                                                                                              |
| G | <Gain>      | 1<br>2<br>Q                         | <Motor Gain><br><Position Gain><br>Quit                                                                                                       | (JOINT) / Quit<br>F1, F2, F3, F4, Quit                                                                                                                                                                                                |
| L | <Limit>     | 1<br>2<br>3<br>4<br>5<br>6<br><br>Q | <Max Voltage><br><Acceleration Limit><br><Velocity Limit><br>< $\omega$ Limit><br>< $\omega_{\text{dot}}$ Limit><br><Angle Limit><br><br>Quit | (JOINT) / Quit<br>F1, F2, F3, F4, Quit<br>F1, F2, F3, F4, Quit<br>(JOINT) / Quit<br>(JOINT) / Quit<br>1 脚1の第一関節<br>2 脚2の第一関節<br>3 脚3の第一関節<br>4 脚4の第一関節<br>5 脚1～4の第二関節<br>6 脚1～4の第三関節<br>7 脚1～4の第三関節(相対角度)<br>8 ポテンショメータ可動範囲<br>Q Quit |
| S | <Sampling>  | 1<br>2<br>3<br>4<br>Q               | Sampling time<br>Log buffer size<br>Sampling skip<br>強制書き出し&初期化<br>Quit                                                                       |                                                                                                                                                                                                                                       |
| D | <Directory> | 1<br>2<br>3<br>Q                    | Tips<br>Log<br>Ini<br>Quit                                                                                                                    | Tip fileのデフォルトパス<br>Log (TIT*.LOG) のデフォルトパス<br>初期設定ファイル (titan.ini/tit org.ini)                                                                                                                                                       |
| Q | Quit        |                                     |                                                                                                                                               |                                                                                                                                                                                                                                       |

表 4-2 メニュー項目 (パラメータ)

(JOINT)は , キー入力にて 2 桁の JOINT 番号(脚番号 + 関節番号)で選択の後に , 数値をキー入力する . 指定範囲外 ( 1 1 ~ 4 3 ではない ) の場合は無視する .

## 5 各種ファイルの書式

### 5 - 1 原点記憶ファイル(tit\_org.ini)

```
<J11> ¥t <J12> ¥t <J13>  
<J21> ¥t <J22> ¥t <J23>  
<J31> ¥t <J32> ¥t <J33>  
<J41> ¥t <J42> ¥t <J43>
```

空行は無視される。

値はポテンショメータのオフセットを意味し、単位は電圧[V]。従って値が±10[V]の範囲に無かった場合はエラーである。読み込み時に、もし書式の異常があった場合は現在の原点の値を変更しない。

## 6 デバッグ方法

titan.h 中の

```
#if 0  
#define DEBUG  
#endif
```

を

```
#if 1  
#define DEBUG  
#endif
```

とすることで、画面表示が反転（通常は、背景が白）し、デバッガ - の視認性が上がる。

```
make [CR]          : PC-9801 シリーズ + 実際のハードウェア制御用プログラムの生成  
make -fmakeoff      : PC-9801 シリーズ用オフラインモードプログラムの生成  
make -fmakeoffv     : DOS/V 機用オフラインモードプログラムの生成
```

オフラインモードプログラムは、実際のハードウェア（拡張ボードや TitanVIII 本体）無しに「みけ」を実行することができるモードである。実際のロボットが動く代わりに、画面中で TitanVIII のワイヤフレーム像が動くシミュレーションプログラムである。制御ルーチンやユーザー I / F など、全てが実際の物と同一のコードを使用している。ごくごくハードウェア寄りのコードのみを騙すことで実現されている。TIP プログラムのデバッグなどに最適である。TIP プログラムのデバッグには、TIP 実行中に[T]でテキスト情報表示をONにし、[TAB]キーを押すことで画面を切り替

えることで、現在の変数の値(VAR, POS, AGL)、現在実行されているT I Pコードの表示を行うことができる (Trace ON 機能)。

## 7 将来的な展望

### (a) 位置および姿勢のフィードバック

現在、Titan は前進後退しかできないが、将来的には旋回や登坂も可能となる。

姿勢（向き／傾き）は加速度センサにより検出可能である。位置に関しては、QuickMag を用いれば完璧だが、装置が大掛かり過ぎるし、他の用途への使用も多いだろう。P S Dカメラは野口のD Dロボットで用いているが、流用は可能と思われる。

### (b) Windows32bit アプリケーション化

メモリの制限が無くなる、マルチスレッド化、メニューのダイアログ化など、恩恵は多い。ランプ（現在は1 2個まで）、スイッチ（キー入力）のデザインの改善、多機能化も期待できる。

しかし Windows 化(Visual C++)を行うことでソースリストの可読性は一気に落ちてしまい、後進による改造が不可能になる恐れもある。MS-DOS (Borland C) → Windows(Visual C++)化が容易に行える（ユニット化）ことが条件である。

### (c) 似非ダイナミクス

OFFLINE モードにおける TITAN の表示に、簡単なダイナミクスを入れる。要するに「転びそうか否か」を姿勢の変化によって表現する。

### (d) ノイズフィルタ

村上製作のノイズフィルタを導入する。その為には、まず彼のノイズフィルタプログラムを汎用化し、必要なルーチンのみを組み込む。

### (e) 過電圧、タイマーオーバーライドの表示

現在は MAX\_VOLT の設定値を越えたか否かは全く外部からは知り得ない。またタイマーがオーバーライドされた場合には非常停止してしまう。これをラッチ式のランプにしてオペレータに提示する。ランプの消灯は意図的に非常停止を発生させ、非常停止クリアする。