

顔画像合成モジュールコマンド一覧

四倉 達夫

プロトコル Version 2.0.6 (2009/02/13)

1. 基本パラメータの直接制御

1.1. Face

動作 set

表情パラメータ値を直接設定する．表情パラメータ ID としては，制御対象の表情パラメータ ID を指定する．パラメータ値としては，制御対象の変化量[%]を指定する．

set Face.表情パラメータ ID = パラメータ値

属性 Face.1：表情パラメータ ID=1（他のパラメータ ID も数字指定）

表情パラメータ値：（整数）最小 1，最大 36

パラメータ値：（整数）最小 0，最大 200

- ・ set Face.1 = 70
 - 表情パラメータ ID1 のパラメータを 70%に設定する．つまり，70%の変形率で変形する

1.2. Mouth

動作 set

口形状パラメータを直接設定する．口形状パラメータ ID としては，制御対象の口形状パラメータ ID を指定する．パラメータ値としては，制御対象の変化量[%]を指定する．

set Mouth.口形状パラメータ ID = パラメータ値

属性 Mouth.1：口形状パラメータ ID=1（他のパラメータ ID を数字指定）

口形状パラメータ値：（整数）最小 1，最大 17

パラメータ値：（整数）最小 0，最大 200

- ・ set Mouth.1 = 70
 - 口形状パラメータ ID=1 のパラメータを 70%に設定する．つまり，70%の変化率で変形する．

1.3. HeadRotRel

動作 set

現在の位置を基準として、頭部を回転させる。回転軸 ID は制御対象の回転軸を整数で指定する。各方向回転量は[degree]を指定する。

set HeadRotRel.回転軸 ID = x 方向回転量 y 方向回転量 z 方向回転量

属性 HeadRotRel.1 : 回転軸 ID=1 (他の回転軸 ID も数字指定)

回転軸 ID : (整数) 設定可能 ID 数 1

x, y, z 方向回転量 : (Double 型, 浮動小数点)

- set = HeadRotRel.1 = 30 40 50
 - 回転軸 1 を中心に, 現在の位置から x 軸方向に 30[degree], y 軸方向に 40[degree], z 軸方向に 50[degree]頭部を回転する。

1.4. HeadRotAbs

動作 set

正面を基準として、頭部を回転させる。回転軸 ID は制御対象の回転軸を整数で指定する。各方向回転量は[degree]を指定する。

set HeadRotAbs.回転軸 ID = x 方向回転量 y 方向回転量 z 方向回転量

属性 HeadRotAbs.1 : 回転軸 ID=1 (他の回転軸 ID も数字指定)

回転軸 ID : (整数) 設定可能 ID 数 1

x, y, z 方向回転量 : (Double 型, 浮動小数点)

- set = HeadRotAbs.1 = 30 40 50
 - 回転軸 1 を中心に, 正面から x 軸方向に 30[degree], y 軸方向に 40[degree], z 軸方向に 50[degree]頭部を回転する。

1.5. MouthScale

動作 set

発話時の口形状の大きさを直接する。パラメータ値としては、制御対象の変化量[%]を指定する。

set MouthScale = パラメータ値

パラメータ値 : (Double 型, 浮動小数点) 最小 0, 最大 200

- ・ **set MouthScale = 70**
 ➤ 発話時の口形状の大きさを 70%に設定する.

動作 inq

現在の口形状大きさの問い合わせに対して rep により答える.

inq MouthScale

- ・ **inq MouthScale**
 ➤ rep MouthScale = 70 : 現在の口形状の大きさ

1.6. AgentRot

動作 set

正面を基準としてエージェントを回転させる. 各方向回転量[degree]を指定する.

set AgentRot = x 方向回転量 y 方向回転量 z 方向回転量

x, y, z 方向回転量 : (Double 型, 浮動小数点)

- ・ **set AgentRot = 10 20 5**
 ➤ 正面から x 軸方向に 10[degree], y 軸方向に 20[degree], z 軸方向に 5[degree]エージェントを回転する.

1.7. AgentTrans

動作 set

初期位置を基準としてエージェントを移動させる.

set AgentTrans = x 方向移動量 y 方向移動量

x(水平), y (垂直) 方向移動量 : (Double 型, 浮動小数点)

- ・ **set AgentTrans = 1.0 0.2**
 ➤ 正面から x 軸方向に 1.0, y 軸方向に 0.2, エージェントを移動する.

1.8. AgentScale

動作 set

エージェントを拡大・縮小させる.

set AgentScale = スケール量

スケール量 : (Double 型, 浮動小数点) 初期状態を 1.0 とする.

- `set AgentTrans = 1.2`
 - エージェントを正気状態より 1.2 倍する.

1.9. AgentEnable

動作 `set`

エージェントの表示の有無を決定する.

`set AgentEnable = コマンド`

コマンド: ENABLE, DISABLE の 2 種類用意されている.

- `set AgentEnable = ENABLE`
 - エージェントを表示させる
- `set AgentEnable = DISABLE`
 - エージェントを画面上から消す

1.10. AgentAlpha

動作 `set`

エージェントの透明度を指定する.

`set AgentAlpha = 透明度`

透明度: (Double 型、範囲 0.0[非表示]から 1.0[表示]まで)

- `set AgentAlpha = 0.5`
 - エージェントを半透明化する
- `set AgentAlpha = 0.0`
 - エージェントを透明化する (画面上から消す)

但し、AgentAlpha コマンドと AgentEnable コマンドは独立して動作する。そのため、AgentEnable を非表示、AgentAlpha を 1.0 としても表示されない。また AgentEnable を表示、AgentAlpha を 0.0 としても表示されない。

2. 基本動作, 基本表情

2.1. FaceMot

動作 set

指定した動作ID, 動作パターンIDの基本動作を行う. 動作IDはBLINK, NOD, REDUSE, LISTEN, USERDEFの5種類. 動作パターンIDは整数値で指定する.

set FaceMot = 動作ID パターンID

動作ID: BLINK (瞬き), NOD (頷き), REFUSE (首を横に振る), LISTEN (話を聞いているように首を傾げる), USERDEF (ユーザ独自定義) の5種類

パターンID: (整数) 最小1, 最大....各IDで異なる.

- set FaceMot = BLINK 1
 - 動作パターン1に指定された「瞬き」を行う.

2.2. FaceExp

動作 set

指定された表情ID, パターンIDの基本表情を指定の頻度, 継続長で行う. 表情IDはHAPPY, DISGUSTED, SAD, ANGRY, SURPRISED, FEARED, USERDEF, NATURALの8種類. 動作パターンIDは整数値で指定する. 表情の強度は変化量[%]を指定する. 継続長は表情の継続長[ms]を指定する. 継続長が0のときは, 次の表情変形指令を受けるまで指定の表情を継続する. NATURALは通常の顔画像に戻すためのIDでパターンID, 強度, 継続長の指定は必要ない.

set Face = 表情ID パターンID 強度 継続長

表情ID: HAPPY (喜び), DISGUSTED (嫌悪), SAD (悲しみ), ANGRY (怒り), SURPRISED (驚き), FEARED (恐れ), USERDEF (ユーザ独自定義), NEUTRAL (初期状態) の8種類

パターンID: (整数) 現在4パターン (パターン No.2-5) のみ定義可能.

強度: (整数) 最小0, 最大200

継続長: (整数) 最小1, 最大の制限はなし, 但し0を指定した時は次の表情変形指令を受けるまで指定の表情を継続する.

- set FaceExp = HAPPY 1 100 1000
 - 動作パターン1に指定された「喜び」の表情を強度100[%], 継続長1000[ms]で行う.

- `set FaceExp = SAD 1 70 0`
 - 動作パターン 1 に指定された「悲しみ」の表情を強度 70[%], 次の表情変形指令を受けるまで継続する.
- `set FaceExp = NEUTRAL`
 - 顔画像の表情を元の状態に戻す.

2.3. EyeRot

動作 set

正面を基準として, 眼球 (両目) を回転させる. 各方向回転量は[degree]を指定し, 回転方向は x 軸, y 軸回転を用意している. また継続時間は眼球の移動時間[ms]で指定する. 継続長が 0 の時は, 次の眼球移動の指令を受けるまで指定の動きを継続する.

set EyeRot = x 方向回転量 y 方向回転量 継続長

x, y 方向回転量: (Double 型, 浮動小数点)

継続長: (整数) 最小 1, 最大の制限はなし, 但し 0 を指定したときは次の眼球運動の指令を受けるまで指定の動きを継続する.

- `set EyeRot = 20.0 0.0 1000`
 - 正面から x 軸方向に 20.0[degree], y 軸方向に 0.0[degree]継続長 1000[ms]で眼球 (両目) の移動を行う.
- `set EyeRot = -10.0 20.0 0`
 - 正面から x 軸方向に -10[degree], y 軸方向に 20.0[degree], 次の表情変形指令を受けるまで継続する.

2.4. EyeRotRight

動作 set

正面を基準として, 右眼球を回転させる. 各方向回転量は[degree]を指定し, 回転方向は x 軸, y 軸回転を用意している. また継続時間は眼球の移動時間[ms]で指定する. 継続長が 0 の時は, 次の眼球移動の指令を受けるまで指定の動きを継続する.

set EyeRotRight = x 方向回転量 y 方向回転量 継続長

x, y 方向回転量: (Double 型, 浮動小数点)

継続長: (整数) 最小 1, 最大の制限はなし, 但し 0 を指定したときは次の眼球運動の指令を受けるまで指定の動きを継続する.

- `set EyeRotRight = 20.0 0.0 1000`

- 正面から x 軸方向に 20.0[degree], y 軸方向に 0.0[degree], 継続長 1000[ms]で右眼球の移動を行う.
- `set EyeRotRight = -10.0 20.0 0`
 - 正面から x 軸方向に-10[degree], y 軸方向に 20.0[degree], 次の表情変形指令を受けるまで,右眼球の移動状態を継続する.

2.5. EyeRotLeft

動作 set

正面を基準として, 左眼球を回転させる. 各方向回転量は[degree]を指定し, 回転方向は x 軸, y 軸回転を用意している. また継続時間は眼球の移動時間[ms]で指定する. 継続長が 0 の時は, 次の眼球移動の指令を受けるまで指定の動きを継続する.

set EyeRotLeft = x 方向回転量 y 方向回転量 継続長

x, y 方向回転量: (Double 型, 浮動小数点)

継続長: (整数) 最小 1, 最大の制限はなし, 但し 0 を指定したときは次の眼球運動の指令を受けるまで指定の動きを継続する.

- `set EyeRotLeft = 20.0 0.0 1000`
 - 正面から x 軸方向に 20.0[degree], y 軸方向に 0.0[degree]継続長 1000[ms]で左眼球の移動を行う.
- `set EyeRotLeft = -10.0 20.0 0`
 - 正面から x 軸方向に-10[degree], y 軸方向に 20.0[degree], 次の表情変形指令を受けるまで左眼球の移動状態を継続する.

3. リップシンク

3.1. LipSync

動作 set

音声合成と同期した口形状アニメーションの準備を行う.

`set Lipsync.属性名 = パラメータ`

利用可能属性名: pho のみ

set LipSync.pho = 音素 1 音素継続長 1 音素 2 音素継続長 2 音素 3 音素継続長 3...

属性 LipSync.pho: 属性名=pho

音素系列と音素継続長[ms]を指定する. 値は音声合成モジュールの Text.pho の値を利用する.

- `set LipSync.pho = h 80 a 100 i 200`
 - 発話文章の音素系列 “h, a, i” および音素継続時間を設定する.

動作 inq

用意した口形状アニメーションの音素系列, 音素継続時間, 総時間の問い合わせに対して `rep` により答える.

`inq LipSync`.属性名

使用可能属性名: `pho`, `dur`, 属性なし

`inq LipSync.pho`

属性 `LipSync.pho`: 属性名=`pho`

現在の設定されている発話音素系列, 音素継続時間[ms]を `rep` により返す.

- `inq LipSync.pho`
 - `rep LipSync.pho = h[80] a[100] i[200]`

`inq LipSync.dur`

属性 `LipSync.dur`: 属性名=`dur`

現在の設定されている文章の総発話時間[ms]を `rep` により返す

- `inq LipSync.dur`
 - `rep LipSync.dur = 380`

`inq LipSync`: 属性なし

現在の設定されている発話音素系列, 音素継続時間, 総発話時間を `rep` により返す.

- `inq LipSync`:
 - `rep LipSync.pho = h[80] a[100] i[200]`
 - `rep LipSync.dur = 380`

3.2. Offset

動作 set

発話タイミングをオフセット値だけずらす (遅らせる).

`set Offset =` オフセット値

オフセット値：発話タイミングをオフセット分遅らせる。単位は[ms]。

- set Offset = 500
 - 発話タイミングを 500[ms]遅らせる。

動作 inq

問い合わせに対して発話タイミングのオフセット値を rep により答える。

inq Offset

- inq Offset
 - rep Offset = 500

3.3. Speak

動作 set

既に LipSync コマンドで準備された口形状アニメーションを決められた時間に開始，もしくは停止する。

set Speak = + 相対時間

相対時間：コマンドを受け取ってから指定した時刻に口形状アニメーションを行う。単位は[ms]。

- set Speak = + 500
 - コマンドを受け取ってから 500[ms]後に口形状アニメーションを行う。

set Speak = コマンド

コマンド：NOW, STOP の 2 種類用意されている。

- set Speak = NOW
 - すぐにあらかじめ準備された口形状アニメーションを行う
- set Speak = STOP
 - 現在の口形状アニメーションの中止を指示する

set Speak.sync = 音声合成のローカル時間 (Ver. 2.0.4-)

属性 Speak.sync：属性名= sync

顔画像合成に音声合成のローカル時間を伝える [msec:整数]

- `set Speak.sync = 1002`
 - 音声合成のローカル時間が1002msecであることを伝える

set Speak.syncoffset = オフセット時間 (Ver. 2.0.4-)

属性 `Speak.syncoffset` : 属性名=`syncoffset`

顔画像合成出力オフセット時間 [msec:整数] ..同期タイミングの微調整用

- `set Speak.syncoffset = 300`
 - オフセット時間を300msecとする (300msec発話開始を遅らせる)

動作 inq

口形状アニメーションに関する問い合わせに対して `rep` によって答える.

`inq Speak.属性名`

使用可能属性名 : `pho`, `dur`, `utt`., `len`, `stat`, 属性なし

inq Speak.pho

属性 `Speak.pho` : 属性名=`pho`

現在の設定されている発話音素系列, 音素継続時間を `rep` により返す.

- `inq Speak.pho`
 - `rep Speak.pho = h [80] a [100] i [200]`

inq Speak.dur

属性 `Speak.dur` : 属性名=`dur`

現在の設定されている文章の総発話時間を `rep` により返す

- `inq Speak.dur`
 - `rep Speak.dur = 380`

inq Speak.utt

属性 `Speak.utt` : 属性名=`utt`

直前に設定されていた発話音素系列, 音素継続時間を `rep` により返す.

- `inq Speak.utt`
 - `rep Speak.utt = a [1000] u [1000] o [1000]`

inq Speak.len

属性 Speak.len : 属性名=len

直線に設定されていた文章の総発話時間を rep により返す

- inq Speak.len
 - rep Speak.len = 3000

inq Speak.text

属性 Speak.text : 属性名=text

現在の設定されている発話音素系列を rep により返す.

- inq Speak.text
 - rep Speak.text = h a i

inq Speak.stat

属性 Speak.stat : 属性名=stat

現在の FSM の発話の処理状態を rep により返す. 処理状態は IDLE (発話文章待ち受け中), PROCESSING (発話文章処理中), READY (文章処理, アニメーション準備完了), SPEAKING (発話アニメーション中) の 4 種類

- inq Speak.stat
 - rep Speak.stat = READY

inq Speak.syncoffset (Ver.2.0.4 -)

属性 Speak.syncoffset : 属性名=syncoffset

現在の発話開始のオフセット時間を問い合わせる

- inq Speak.syncoffset
 - rep Speak.syncoffset = 300

inq Speak : 属性なし

現在の設定されている発話音素系列, 音素継続時間, 総発話時間, 直前に設定されていた発話音素系列, 音素継続時間, 総発話時間, 処理像対を rep により返す. 上記した 4 つの属性を全て実行したときと同様の結果を得る.

- inq Speak

- rep Speak.pho = h [80] a [100] i [200]
- rep Speak.dur = 380
- rep Speak.utt = a [1000] u [1000] o [1000]
- rep Speak.len = 3000
- rep Speak.text = h a i
- rep Speak.stat = READY

動作 prop

口形状アニメーション出力に関する属性値の出力仕様を設定する.

prop Speak.属性名

使用可能属性名 : pho, dur, utt, len, stat, 属性なし

prop Speak.pho

属性 Speak.pho : 属性名=pho

音素系列と音素継続長の出力仕様を設定する

- prop Speak.pho = AutoOutput
 - 現在の設定されている発話文章の音素系列を変更された時点で rep により返すように設定する.
- prop Speak.pho = NoAutoOutput
 - 現在の設定されている発話文章の音素系列と音素継続時間を inq にて問い合わせるときに rep により返すように設定する.

prop Speak.dur

属性 Speak.dur : 属性名=dur

総発話時間の出力仕様を設定する

- prop Speak.dur = AutoOutput
 - 現在の設定されている発話文章の総発話時間を変更された時点で自動的に rep により返すように設定する.
- prop Speak.dur = NoAutoOutput
 - 現在の設定されている発話文章の総発話時間を inq にて問い合わせたときに rep により返すように設定する.

prop Speak.utt

属性 Speak.utt : 属性名=utt

既に発話が完了した音素系列と音素継続長の出力仕様を設定する

- `prop Speak.utt = AutoOutput`
 - 既に発話が完了した音素系列と音素継続長を変更された時点で自動的に `rep` により返すよう設定する.
- `prop Speak.utt = NoAutoOutput`
 - 既に発話が終了した音素系列を `inq` にて問い合わせた時に `rep` により返すように設定する.

prop Speak.len

属性 `Speak.len` : 属性名 `len`

既に発話が完了した発話時間の出力仕様を設定する

- `prop Speak.len = AutoOutput`
 - 既に発話が終了した発話時間に変更された時点で自動的に `rep` により返すように設定する.
- `prop Speak.len = NoAutoOutput`
 - 既に発話が終了した発話時間を `inq` にて問い合わせた時に `rep` により返すように設定する.

prop Speak.stat

属性 `Speak.stat` : 属性名 `stat`

現在の発話アニメーション処理の状態の出力仕様を設定する

- `prop Speak.stat = AutoOutput`
 - 現在の発話アニメーション処理の状態を変更された時点で自動的に `rep` により返すように設定する.
- `prop Speak.stat = NoAutoOutput`
 - 現在の発話アニメーション処理の状態を `inq` にて問い合わせた時に `rep` により返すように設定する.

prop Speak : 属性名なし

現在の口形状アニメーションの出力に関する全ての属性値の出力仕様を設定する.

- `prop Speak := AutoOutput`
 - 現在の `Speak` に設定されている全情報を変更された時点で自動的に `rep` により返

すように設定する.

- `prop Speak.pho = AutoOutput`
- `prop Speak.dur = AutoOutput`
- `prop Speak.utt = AutoOutput`
- `prop Speak.len = AutoOutput`
- `prop Speak.stat = AutoOutput` と同様
- `prop Speak = NoAutoOutput`
 - 現在の `Speak` に設定されている全情報を `inq` にて問い合わせた時に `rep` により返すように設定する.
 - `prop Speak.pho = NoAutoOutput`
 - `prop Speak.dur = NoAutoOutput`
 - `prop Speak.utt = NoAutoOutput`
 - `prop Speak.len = NoAutoOutput`
 - `prop Speak.stat = NoAutoOutput` と同様

4. 顔画像の選択・保存

4.1. MaskSet

動作 `inq`

使用可能な顔画像の問い合わせに対して `rep` により答える.

`inq MaskSet`

- `inq MaskSet`
 - `rep MaskSet = "man01, man02, woman01"` : 利用可能な顔画像 ID 列

4.2. Mask

動作 `set`

顔画像生成に使用する顔画像を設定する.

`set Mask = 顔画像 ID`

顔画像 ID : (文字列) あらかじめ登録した ID を入力

- `set Mask = man01`
 - 顔画像生成に使用する `man01` (顔画像 ID) の指定をする. 顔画像 ID は `MaskSet`

で問い合わせた顔画像 ID を指定する.

4.3. NowMask

動作 **inq**

現在使用している顔画像を **rep** にて答える.

inq NowMask

- **inq NowMask**
 - **rep NowMask** = “man01” : 現在の顔画像 ID

4.4. SaveImage

動作 **set**

現在出力している顔画像の保存

set SaveImage = ファイル名

ファイル名 : 保存先の指定, 保存画像形式は Windows Bitmap 形式 (拡張子.bmp), もしくは SGI (RGB) 形式(拡張子.rgb)で保存可能. これら形式は記述したファイル名の拡張子で判断する. また, 他のフォルダに保存したい場合はフォルダ (ディレクトリ) 名も明記すること.

- **set SaveImage = ../snapshot/hap.rgb** : ../snapshot/ディレクトリ内に hap.rgb (SGI 形式) にて顔画像を保存.
- **set SaveImage = ../snapshot/sad.bmp** : ../snapshot/ディレクトリ内に sad.bmp (bitmap 形式) にて顔画像を保存
-

4.5. Capture

動作 **set**

set Capture = フォルダ名 ファイル名 描画フレームレート (Ver.2.0.5-)

画面のキャプチャ (を行う).

フォルダ名 : 保存先ディレクトリ名を入れる (例 : test ディレクトリ、カレントは. (ドット)).

ファイル名 : 連番ファイル名(例 : ファイル名を test と入力すると、test00000.bmp から順番に保存されます)

描画フレームレート : 描画フレームレート (FPS) を設定します (例 : 30fps 秒 30 コマで保存、一秒間に 30 枚画像が作成されます).

画面キャプチャは **set Speak = NOW** が実行されたときからキャプチャを開始し、発話終了後にキャプチャ終了、キャプチャモードを抜け、通常（リアルタイム描画モード）に戻ります。

- ・ **set Capture = . sample_ 30 :** ./ディレクトリ内に **sample_XXXXX.bmp** (X は数字が入ります) ファイルを保存。

動作 inq

キャプチャ可能かどうかの問い合わせに対して **rep** により答える。

inq Capture

- ・ **inq Capture**
 - **rep BackgroundSet = ENABLE or DISABLE**

5. 自律動作

5.1. Autonomous

動作 set

指定された動作 ID, パターン ID の自律動作の設定を行う。

set Autonomous = 動作 ID パターン ID

動作 ID : BLINK (瞬き), MOVE (頭部の動き), USERDEF (ユーザ独自定義) の 3 種類用意してある。

パターン ID: (整数) 現在, BLINK はパターン 1~6, MOVE はパターン 1 のみ, USERDEF はパターン 1 のみ用意する。パターン ID が 0 の場合, いずれの動作 ID でも指定されていた自律動作が停止する。

- ・ **set Autonomous = BLINK 3**
 - 動作パターン 3 に指定された「瞬き」の自律動作を設定する。
- ・ **set Autonomous = MOVE 1**
 - 動作パターン 1 に指定された「頭部の動き」の自律動作を設定する
- ・ **set Autonomous = USERDEF 0**
 - 「ユーザ定義」の自律動作を停止する。

6. 背景・描画モード

6.1. Background

動作 **set**

背景の設定. 指定された背景 ID を顔画像の背景として登録する.

set Background = 背景 ID

背景 ID : (文字列) あらかじめ登録した ID を入力

- **set Background = bg1**
 - 背景に使用する bg1 (背景 ID) の指定をする. 背景 ID は BackgroundSet で問い合わせた背景 ID を指定する.

6.2. BackgroundSet

動作 **inq**

使用可能な背景の問い合わせに対して **rep** により答える.

inq BackgroundSet

- **inq BackgroundSet**
 - **rep BackgroundSet = "bg1,bg2"** : 利用可能な背景 ID 列

6.3. NowBackground

動作 **inq**

現在使用している背景を **rep** にて答える.

inq NowBackground

- **inq NowBackground**
 - **rep Background = "bg1"**

6.4. ViewMode

動作 **set**

顔画像の表示モードの設定.

set ViewMode = 表示 ID

表示 ID : TEXTURE (テクスチャ表示), TEXTURE_WITH_WIREFRAME (テクスチャ + ワイヤフレーム表示), WIREFRAME (ワイヤフレーム表示) の 3 種類用意してある。また, WIREFEAME モードでは背景の表示は行われない。

- set ViewMode = TEXTURE
 - 顔画像の表示モードをテクスチャ表示にする

7. モジュールの起動・終了・問い合わせ

7.1. Run

動作 set

モジュールの動作を指定する。

set Run = パラメータ

パラメータ : INIT (初期設定状態), EXIT (モジュールの終了) の 2 種類存在する。

- set Run = INIT
 - モジュールを居気設定状態にする。
- set Run = EXIT
 - モジュールを終了する。

動作 inq

モジュールの動作状態を問い合わせに対して rep により答える。ただし rep に返すパラメータは LIVE のみ (動作されていない場合は、コマンドを受け付けない)。

inq Run

- inq Run
 - rep Run = LIVE : モジュールが処理開始可能状態である時の出力

7.2. ProtocolVersion

動作 inq

モジュールの準拠している標準通信プロトコルの Version 情報の問い合わせに対して rep により答える。

inq ProtocolVersion

- inq ProtocolVersion
 - rep ProtocolVersion = “Protocol Ver. 2.0.2” : モジュールのプロトコル Version 情報

7.3. ModuleVersion

動作 inq

モジュール自体の Version 情報の問い合わせに対して rep により答える.

inq ModuleVersion

- inq ModuleVersion
 - rep ModuleVersion = “Module Ver. 2.2.0” : モジュールの Version 情報

コマンド一覧表 Version.2.0.6

1. 基本パラメータの直接制御	
パラメータ名	機能
Face	表情パラメータ制御コマンドの直接制御
Mouth	口形状パラメータ制御コマンドの直接制御
HeadRotRel	現在の位置を基準とした頭部の回転
HeadRotAbs	正面を基準とした頭部の回転
MouthScale	口形状の大きさを設定
AgentRot	エージェントの回転
AgentTrans	エージェントの移動
AgentScale	エージェントのスケール
AgentEnable	エージェントの表示の有無
AgentAlpha	エージェントの透明度

2. 基本動作, 基本表情	
パラメータ名	機能
FaceMot	動作パターン ID に指定された基本動作の制御
FaceExp	強度と継続長の指定された基本表情の制御
EyeRot	両目の動きを制御
EyeRotRight	右目の動きを制御
EyeRotLeft	左目の動きを制御

3. リップシンク	
パラメータ名	機能
LipSync	音声合成との同期準備
Offset	発話開始時間をずらす
Speak	決められた時刻に発声

4. 顔画像の選択・保存	
パラメータ名	機能
MaskSet	使用可能な顔画像の問い合わせ
Mask	顔画像生成に使用する顔画像の選択
NowMask	現在の顔画像の問い合わせ
SaveImage	現在出力している顔画像の保存
Capture	画面のキャプチャ

5. 自律動作	
パラメータ名	機能
Autonomous	自律動作の設定

6. 背景・描画モード	
パラメータ名	機能
Background	背景の設定
BackgroundSet	使用可能な背景の問い合わせ
NowBackground	現在の背景の問い合わせ
ViewMode	描画モード(Wireframe,Texture 等)の設定, 問い合わせ

7. モジュールの起動・終了・問い合わせ	
	機能
Run	モジュールの起動, 再起動, 停止の設定, および状態の問い合わせ
ProtocolVersion	モジュールの準拠している標準通信プロトコル version の問い合わせ
ModuleVersion	モジュール自身の version 情報の問い合わせ