

NEDO 特別講座
市場化成果活用コース
モバイルマニピュレーション

⑤ SEED Platform Robots



アプリケーションの開発に専念できる 手足の付いたUSB機器

組合わせ可能

簡単に動作が作成可能

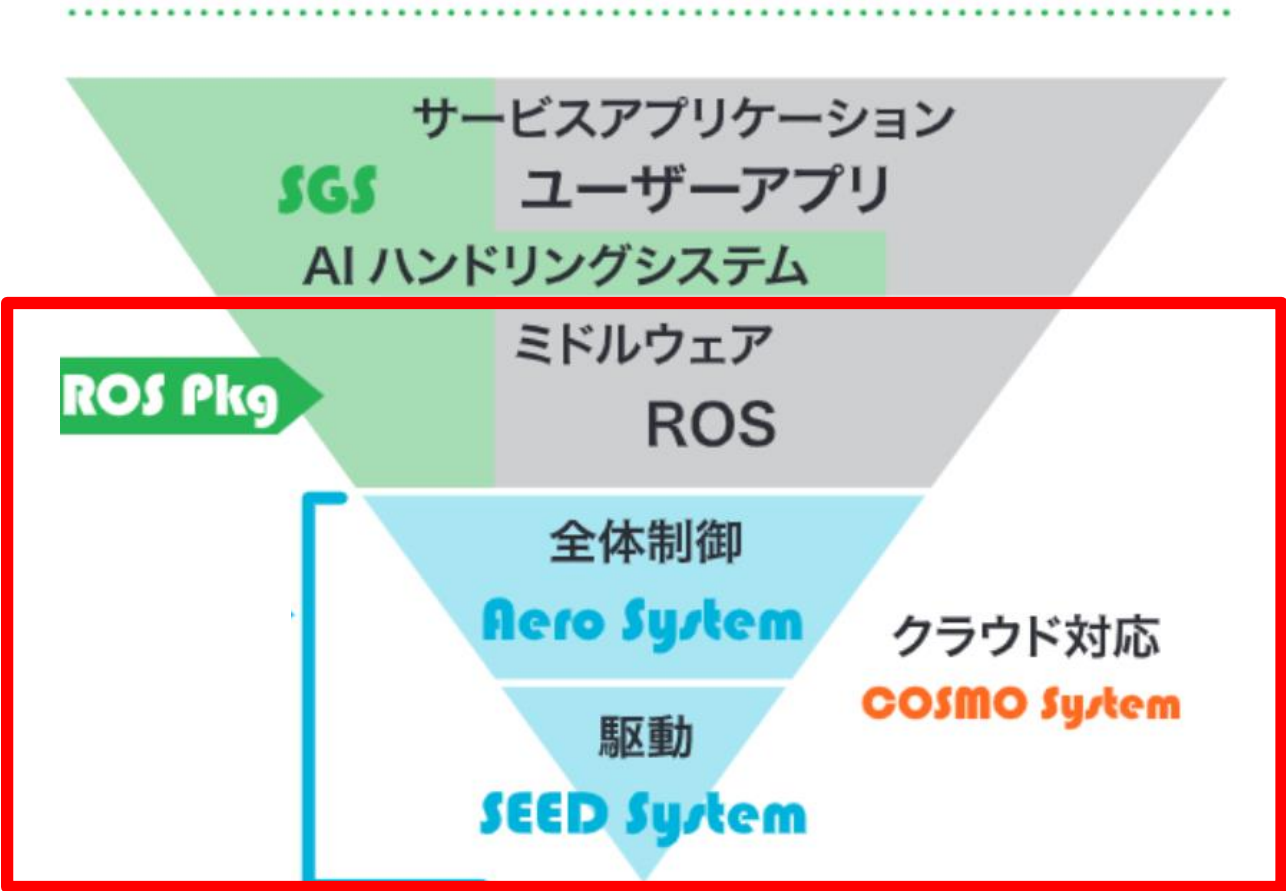
カスタマイズ可能

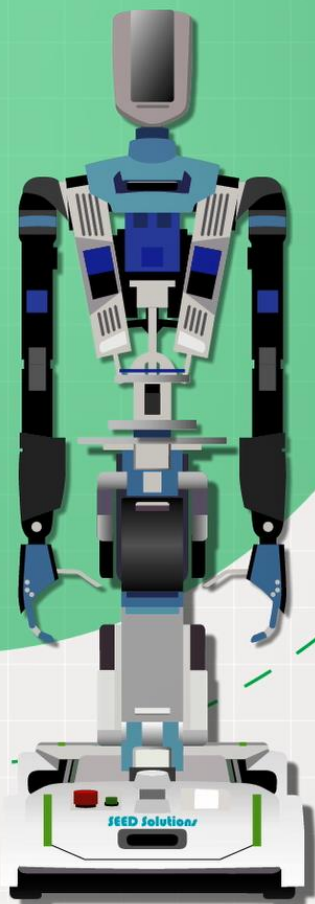
ROS対応

外装デザインが自由

動作信頼性







SEED Platform Robots

サンプルソフト

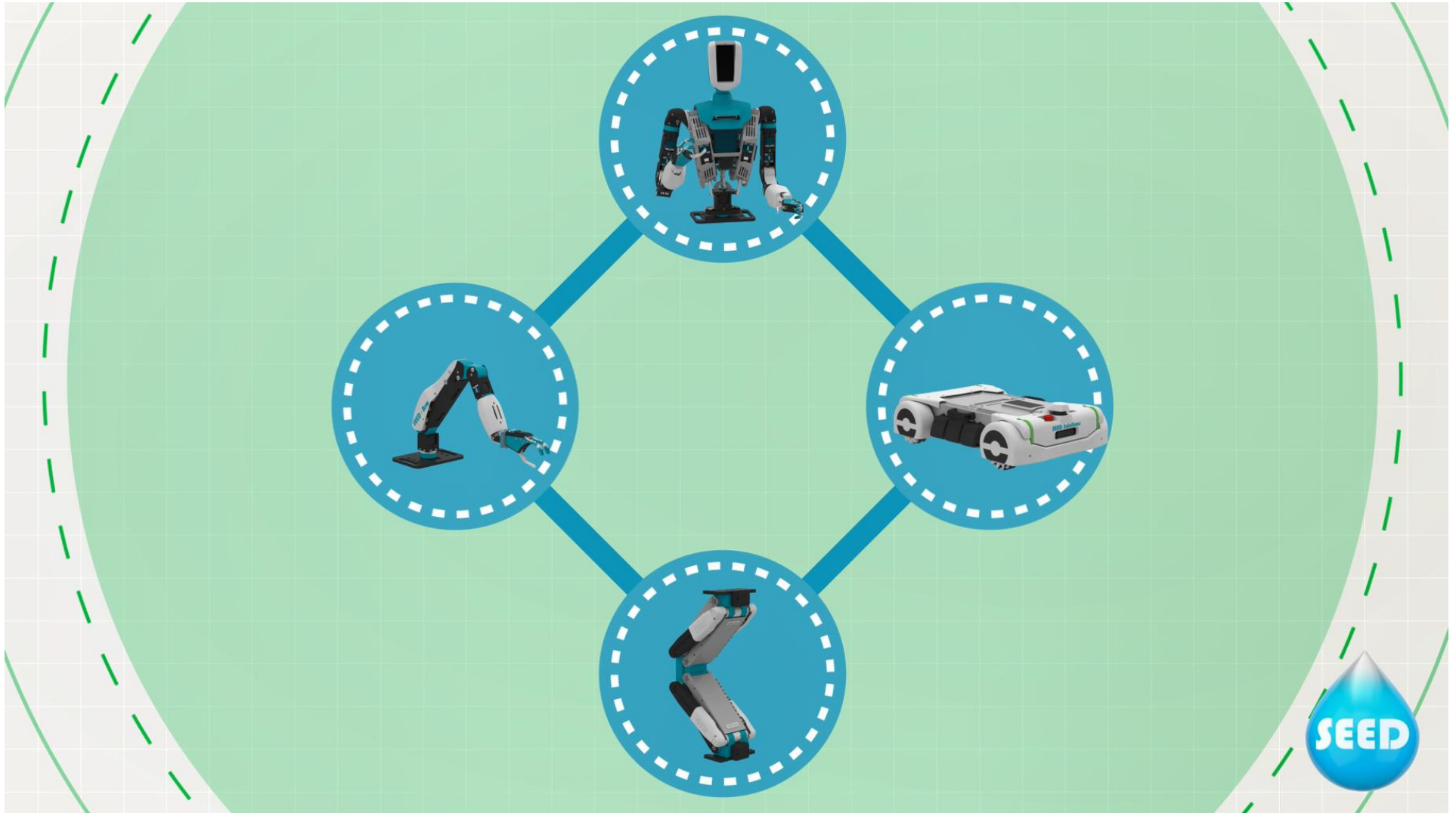
チュートリアル

アプリケーション開発

外観自由

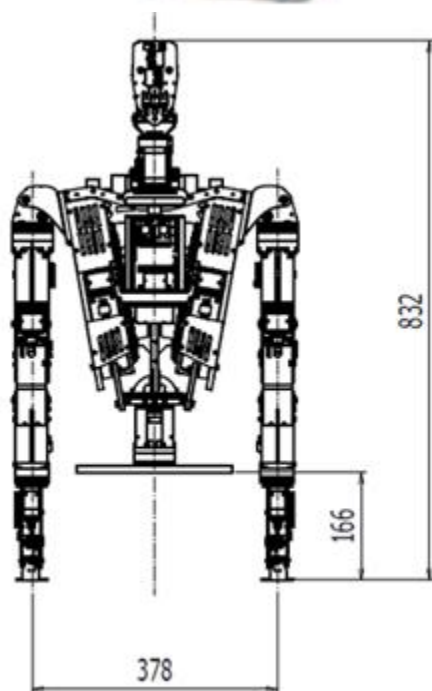
アプリケーションの開発に注力し、最短で自由な外観を持ったサービスロボットが構築できる

Platform Robots 種別



4種類のPlatform Robotsが製品化されています。





◎20自由度の人型ロボット

腕 7軸x2

腰 3軸

頭 3軸

◎可搬重量

2Kg

TRX装着時 1.5Kg

TRX回転軸追加可能

◎繰り返し位置決め精度

±0.5mm

◎アンダーウェア 冷却機能



◎6自由度の片腕ロボット

◎可搬重量

2Kg

TRX装着時 1.5Kg

TRX回転軸追加可能

◎繰り返し位置決め精度

±0.5mm

SEED-lifter 概要



◎昇降・前後 移動

→平行移動

→上下最大 500mm

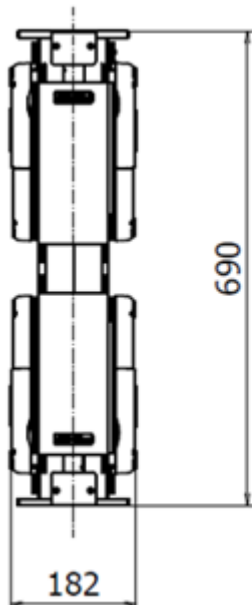
→前後最大 250mm

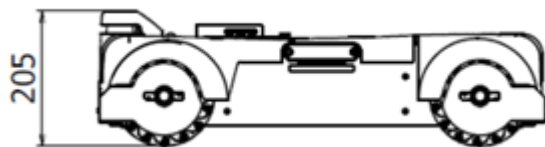
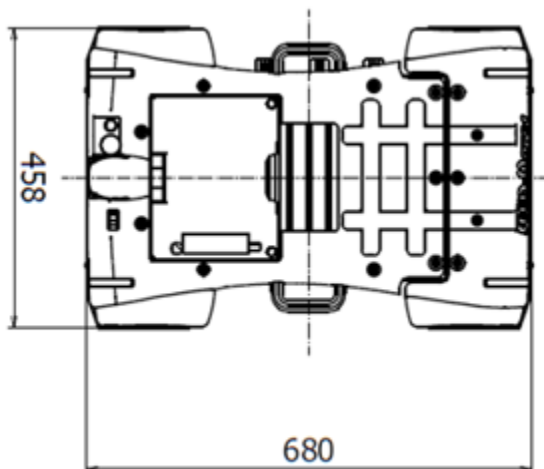
※上下高さによる

◎SPEC

・可搬重量 20Kg

・消費電力 20W





◎全方位移動

4輪 Mecanumホイール

◎SLAM機能

- ・LIDER
- ・DepthCAM 前後

◎電源機能

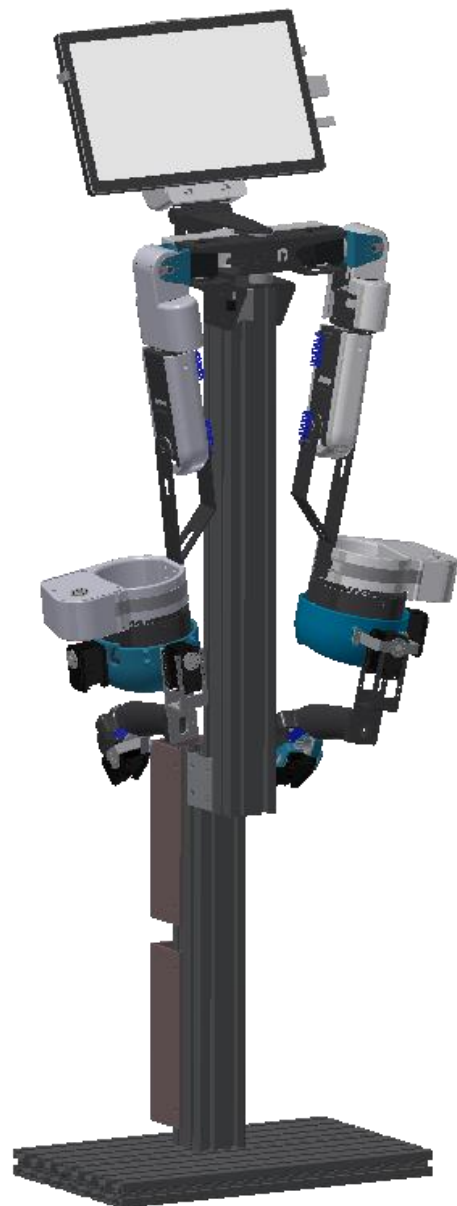
- ・バッテリー 1kW
- ・外部電源供給
- ・充電機能 (Option)

◎LED表示

◎バンパー機能

◎SPEC

- ・可搬重量 35Kg
- ・最大速度 0.5m/sec
- ・重量 30kg ※バッテリー込み



◎15自由度の人型トレーサー

腕 7軸 x2

腰 1軸

◎一体型

PC

モニター

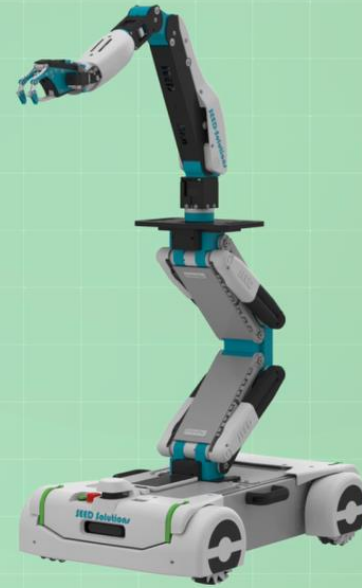
モバイルルーター

Tracerコントローラー

Platform Robots



SEED-Noid-Mover

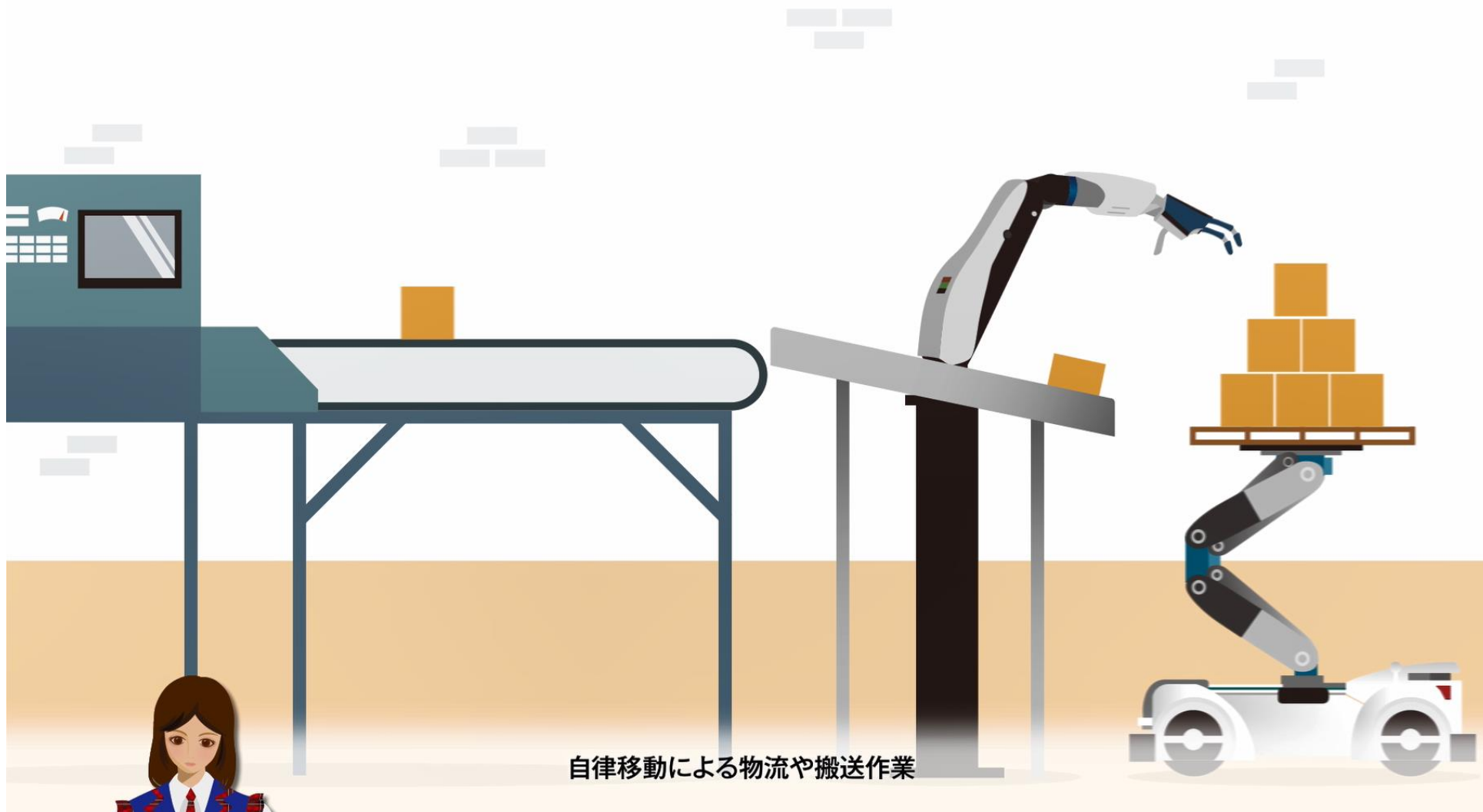


SEED-Arm-Mover



人型のモバイルマニピュレーターと、単腕のモバイルマニピュレーターを構築可能

Platform Robots使用例1



自律移動による物流や搬送作業



さまざまな 外観を構築し遠隔操作によるコミュニケーション

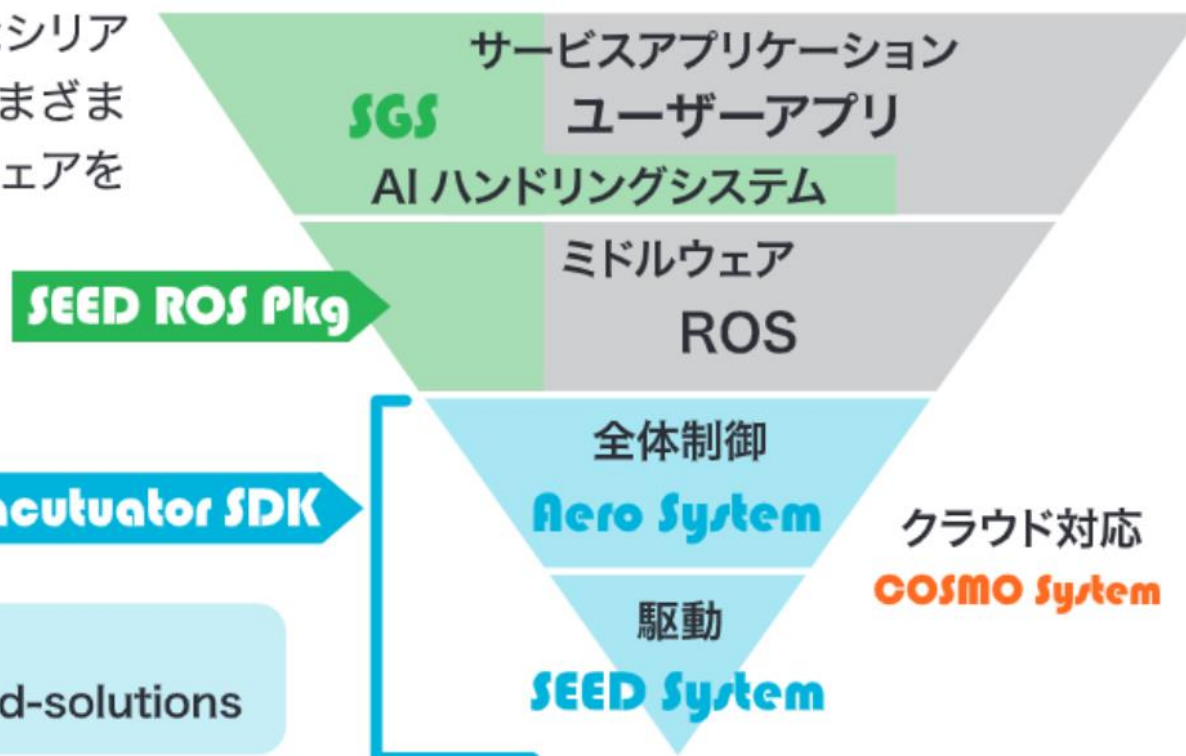
Platform Robots使用例3



サービス産業における 店頭業務まで

Software

USBデバイスとして、簡単なシリアルコマンドで動作します。さまざまなレイヤーでの基本ソフトウェアを用意しております。



Github
<https://github.com/seed-solutions>

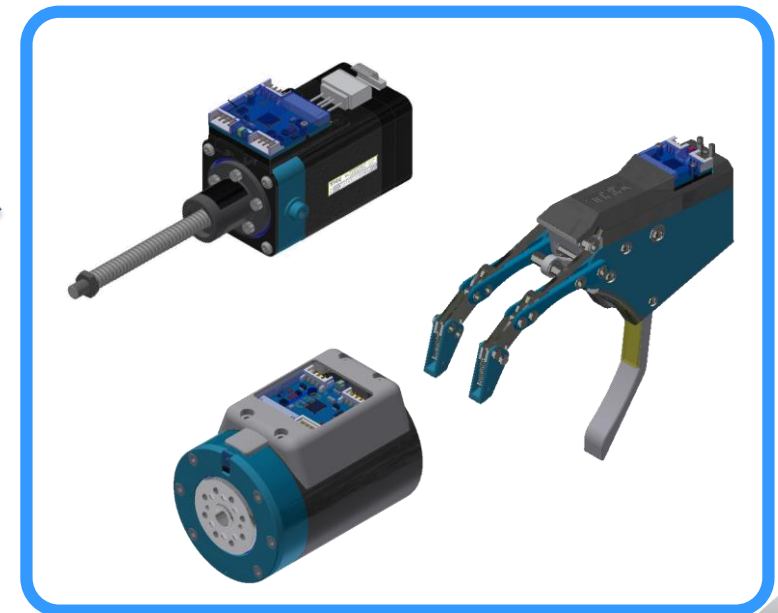
API群

```
//Fixed Parameters-----  
void setTypeNumber(uint8_t_id, const char* _type);  
void setEditorVersion(uint8_t_id, const char* _ver);  
void setMotorCurrentParam(uint8_t_id, uint16_t_driver_max, uint16_t_driver_min);  
void setDummy(uint8_t_id, uint8_t_cmd);  
void setSerialVersion(uint8_t_id, const char* _ver);  
void setMotorAdaptation(uint8_t_id, uint32_t_type, uint16_t_volt, uint16_t_current);  
void setCurrentInstantaneous(uint8_t_id, uint16_t_max, uint16_t_min);  
void setFirmwareVersion(uint8_t_id, const char* _ver);  
void setMotorParam(uint8_t_id, uint8_t_mode, uint8_t_feedback);  
void setEncoderParam(uint8_t_id, uint16_t_encoder_pulse, uint16_t_encoder_resolution);  
  
//Base Parameters-----  
void setIdParam(uint8_t_id, uint8_t_re_id);  
void setEmergencyParam(uint8_t_id, uint8_t_mode, uint8_t_io_no, uint8_t_io_dir);  
void setStopModeParam(uint8_t_id, uint8_t_motor, uint8_t_script);  
void setOperationParam(uint8_t_id, uint8_t_auto_run, uint8_t_script);  
void setOvertravelParam(uint8_t_id, uint8_t_mode, uint8_t_minus, uint8_t_plus);  
void setErrorMotionParam(uint8_t_id, uint8_t_temperature, uint8_t_time);  
void setResponseParam(uint8_t_id, uint8_t_mode);  
void setDioParam(uint8_t_id, uint8_t_io0, uint8_t_io1, uint8_t_io2, uint8_t_io3);  
void setAdParam(uint8_t_id, uint8_t_ad0, uint8_t_ad1, uint8_t_ad2, uint8_t_ad3);  
  
//Motor Settings-----  
void setMotorCurrent(uint8_t_id, uint16_t_max, uint8_t_min, uint8_t_max_speed);  
void setMotorMaxSpeed(uint8_t_id, uint16_t_speed);  
void setMotorControlParameter1(uint8_t_id, uint8_t_back_surge_a, uint8_t_back_surge_b);  
void setMotorControlParameter2(uint8_t_id, uint16_t_i_gain, uint16_t_i_limit);  
void setInPosition(uint8_t_id, uint16_t_value);  
void setACDecelerationRate(uint8_t_id, uint16_t_acc, uint16_t_deceleration);  
void setMotorControlParameter3(uint8_t_id, uint16_t_initial_speed, uint16_t_initial_time);  
void setUpperSoftwareLimit(uint8_t_id, int32_t_limit);  
void setLowerSoftwareLimit(uint8_t_id, int32_t_limit);  
void setMotorRotation(uint8_t_id, uint8_t_pulse_division, uint8_t_pulse_time);  
void setMotorError(uint8_t_id, uint16_t_time, uint32_t_pulse);  
void setMotorErrorLimit(uint8_t_id, uint8_t_temperature, uint8_t_time);  
  
//Script Settings-----  
void setScriptData(uint8_t_id, uint8_t_number, uint8_t_start_line, uint8_t_end_line);  
void writeScriptLine(uint8_t_id, uint8_t_line, const char* _command);
```

Aero Command



SEED Command



ROS

MoveIt!



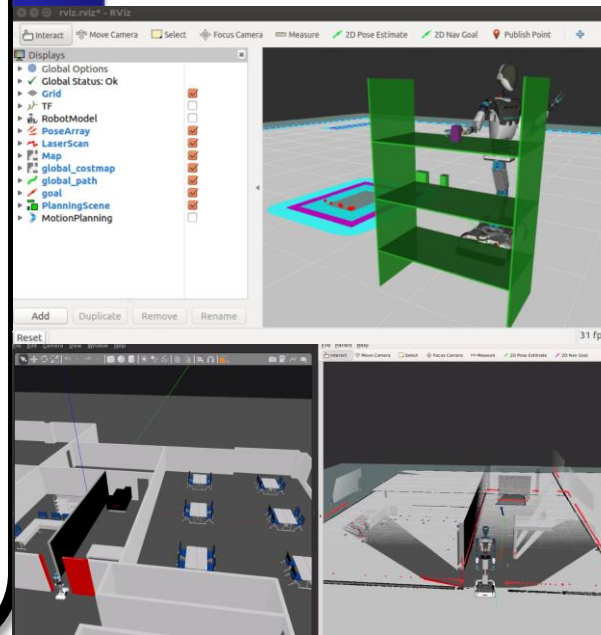
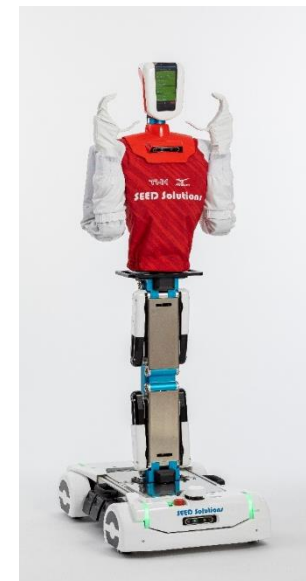
GAZEBO



OpenSLAM
Give your algorithm to the community

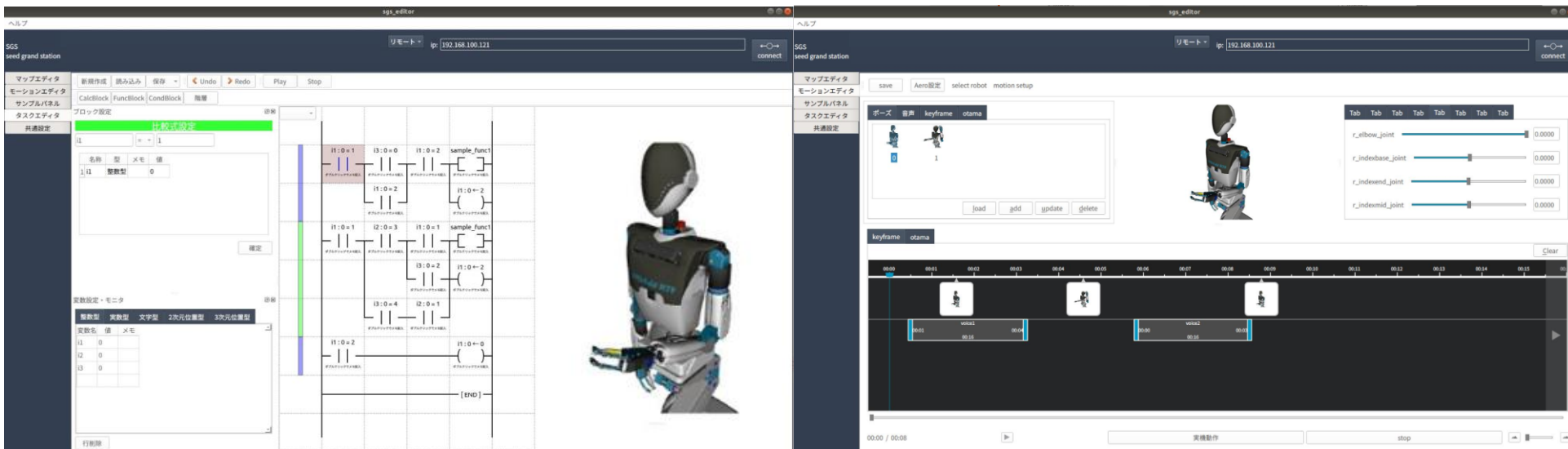
SEED ROS BRIDGE

Aero Command



Platform Robotsへカスタマイズ

パラメーターの設定、Taskの記述、MAPの作成まで、一連の運用まで可能



ラダーライク タスク記述方式

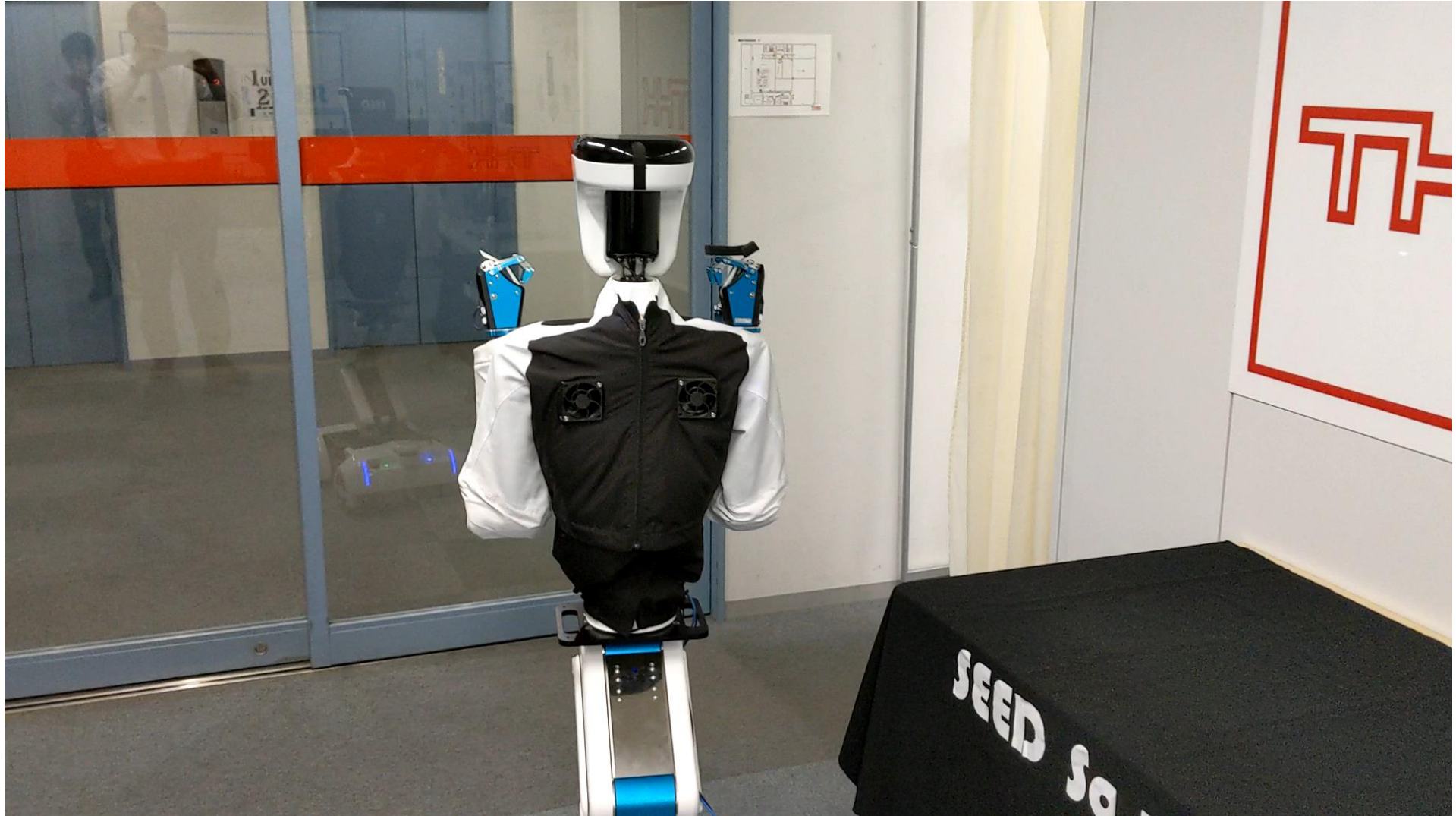
- ROS等を意識しないで、タスクを構築できる
- 設備関連のソフトエンジニアが対応できる
→C等で直接プログラムをしなくて良い
- 内部パラメータが把握しやすい

モーションエディター

- 基本ポーズを組合わせてモーションを作成
- 音にあわせて、動きを調整可能



MOVIE 2



MOVIE 3





ユニット化されたSEEDと、用意されたソフトウェア一群によって、
SMARTにあなただのIDEAをREALIZEする