






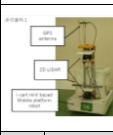
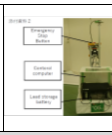













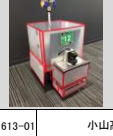







つくばチャレンジ2016 参加チーム一覧

No.	チーム名		1.ハードウェア		2.ソフトウェア		3.環境構築・シミュレータなど		
	ロボット外観	ロボット内観など							
1601	土浦プロジェクト i-Cart Middle		走行部・ボディ	自作	T-frog projectのi-Cart miniをベース。	金体システム	自作	筑波大学知能ロボット研究室のSSMを利用。	自作
			コントローラ	ノートPC 組み込み用PC	Panasonic Let's Note, NVIDIA Jetson TX1, NVIDIA Jetson TK1(2枚)	走行制御系	既製のソフトウェアを改良・改造	筑波大学知能ロボット研究室が公開しているYP-Spurを利用。	
			ドライバ	市販品	ツジ電子 TF-2MD3-R6	センサ処理・認識系	Open-CV その他	GPU用ライブラリ: NVIDIA CUDA 画像幾何変換: Openframeworks 人間・信号認識: Chamer, Caffe	
1602	DEMURA x UNIMO (金沢工業大学) UNIMO AI		走行部・ボディ	市販品を改造	クローラー型パーソナルモビリティ U NMO Grace	金体システム	ROS	ナビゲーション: Moba_base 地図作製: gmapping 自己位置推定: amcl	ROS
			コントローラ	その他	Arduino Mega 2560	走行制御系	ROS	Move_baseを利用。ただし、arduinoに搭載したPIDコントローラは自作。	
			ドライバ	利用した走行部に付随	-	センサ処理・認識系	Open-CV PCL ROS	-	
1603	芝浦工業大学 機械制御工学研究室 やまぶき 6		走行部・ボディ	市販品を改造	セグウェイジャパン株式会社 Black Ship 4Wheels	金体システム	自作	-	その他
			コントローラ	ノートPC	-	走行制御系	自作	-	C言語 (Eclipse) C++言語
			ドライバ	利用した走行部に付随	-	センサ処理・認識系	自作システム Open-CV	-	
1604	YamaneLab (帝京大学) Progress-i MK-II		走行部・ボディ	市販品を改造	T-frogプロジェクトのi-Cart miniをベースに改造。	金体システム	自作	基本的にはC++言語で開発。	自作既製のソフトウェアを改良・改造 ROS
			コントローラ	ノートPC	マウスコンピュータ・LB-C300S-SSD	走行制御系	自作既製のソフトウェアを改良・改造	筑波大学・知能ロボット研究室が開発したyp-spurを中心に、Mobile Robot Programming Toolkit (MRPT) と自作プログラムで構成。	目的に合わせてMRPT, ROS, 自作プログラムを併用
			ドライバ	市販品	T-frogプロジェクト2軸モータドライバTF-2MD3-R6を利用。	センサ処理・認識系	Open-CV ROS	カメラを用いたオブジェクト認識ではOpenCV、障害物検知・回避はMRPT、時々刻々のセンサ情報はArduinoを用いて開発。またその共有として筑波大学・知能ロボット研究室のssm、ログの記録としてROSのrosvizパッケージを利用。それ以外の路面状況認識などは自作。	
1605-01	千葉工業大学工学部未来ロボティクス学科 ロボット設計・制御研究室 ORNE-α		走行部・ボディ	市販品を改造	T-frog Projectが開発したi-Cart mini, i-Cart middleをベースとして開発を行った。	金体システム	ROS	自己位置推定、経路計画、着座人物検出のプログラムの通信として使用。	ROS
			コントローラ	ノートPC	Think pad x-240	走行制御系	ROS	筑波大学知能ロボット研究室が開発した移動ロボットよう走行制御プラットフォームのYP-Spurを使用している。	シミュレーション環境として GAZEBO, 可視化ツールRvizを使用。
			ドライバ	利用した走行部に付随	2軸モータドライバ TF-2MD3-R6を使用した。	センサ処理・認識系	ROS	センサを会社から提供されているROSのパッケージなどを使用。	
1605-02	千葉工業大学工学部未来ロボティクス学科 ロボット設計・制御研究室 ORNE-β		走行部・ボディ	市販品を改造	T-frog Projectが開発したi-Cart mini, i-Cart middleをベースとして開発を行った。	金体システム	ROS	自己位置推定、経路計画、着座人物検出のプログラムの通信として使用。	ROS
			コントローラ	ノートPC	Think pad T-450	走行制御系	ROS	筑波大学知能ロボット研究室が開発した移動ロボットよう走行制御プラットフォームのYP-Spurを使用している。	シミュレーション環境として GAZEBO, 可視化ツールRvizを使用。
			ドライバ	利用した走行部に付随	2軸モータドライバ TF-2MD3-R6を使用した。	センサ処理・認識系	ROS	センサを会社から提供されているROSのパッケージなどを使用。	
1606	宇都宮プロジェクト アロマックス9		走行部・ボディ	自作	-	金体システム	既製のソフトウェアを改良・改造	SSM(筑波大学知能ロボット研究室)を使用、各機能プロセス間でデータ共有。	既製のソフトウェアを改良・改造
			コントローラ	ノートPC 組み込み用PC	AlphaProject社製 SH7136ボード: AP-SH2F-11A	走行制御系	自作	-	eclipseODTを使用
			ドライバ	市販品	Hibot 1-Axis DC Power Module	センサ処理・認識系	Open-CV	データの表示、ビットマップ上の処理に使用。	
1607	チームイエスマン Tsukuba Exploration Rover		走行部・ボディ	市販品を改造	田宮製のラジコンをベースに開発。 Tamiya CR-01 TOYOTA LAND CRUISER 40	金体システム	自作 ROS	各ノードに関しては自作(走行制御部, カメラコントローラ, 画像処理部)。ノード間通信・データ取得のみにROSを利用。	既製のソフトウェアを改良・改造 ROS
			コントローラ	デスクトップPC 組み込み用PC	Intel NUC517RYH Arduino Leonard	走行制御系	自作	走行制御に関しては作りこみに関わらなかったが、自作したライブラリをROSでラップしてノードとして利用した。	開発環境として、エディタにはEclipseを使用。データ取得・リプレイにはROSを使用。
			ドライバ	利用した走行部に付随	モーグドライバ等はラジコンに付随しているものを利用。カメラのドライバには、Video4Linuxを利用。	センサ処理・認識系	Open-CV ROS	画像処理のベースとしては OpenCV を利用。またステレオカメラに関しては、OpenCVより提供されているSGBMをベースとして用いた。	
1608	大阪市立大学 知能情報処理工学研究室 Dulcinea		走行部・ボディ			金体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			

つくばチャレンジ2016 参加チーム一覧

No.	チーム名		1.ハードウェア			2.ソフトウェア			3.準備両者・シミュレータなど
	ロボット外観	ロボット内観など							
1609-01	芝浦工業大学	マイクロメカトロニクス研究室	走行部・ボディ	市販品を改造	T-frog Projectのi-Cart miniをベースにアルミフレームで拡張を施した機体構成となっており、タイヤに関しては土浦プロジェクトのi-Cart middleのものを参考に10インチから12インチに変更している。ロボット全体はプラスチック段ボールとクッションで覆っている。	金体システム	既製のソフトウェアを改良・改造	Streaming data Sharing Manager(ssm)(筑波大学知能ロボット研究室)を用いて複数のプログラムを管理するマルチプロセスを採用している。	自作 既製のソフトウェアを改良・改造
		MML-05-H	コントローラ	ノートPC	Panasonic Let's note CF-SX3	走行制御系	既製のソフトウェアを改良・改造	YP-spur(筑波大学知能ロボット研究室)	ubuntu16.04(Linux)を使用。ロボットの動作プログラムはssmを利用したプロセス間通信によって動いている。
			ドライバ	市販品	T-frogドライバ(T-frog Project)	センサ処理・認識系	自作システム PCL その他	3D LIDARのセンサ情報取得および処理するプログラムを作成。スキャンマッチングを行うためのプログラムとそれを確認するためのviewerを作成。	
1609-02	芝浦工業大学	マイクロメカトロニクス研究室	走行部・ボディ	市販品を改造	T-frog Projectのi-cart miniをベースに開発した。土浦プロジェクトのi-cart middleを参考にタイヤ径を変更した。	金体システム	既製のソフトウェアを改良・改造	筑波大学知能ロボット研究室のStreaming Data Sharing Managerを使用してプログラム開発を行った。	自作 既製のソフトウェアを改良・改造
		MML-05-Y	コントローラ	ノートPC	Panasonic Let's note CF-LX5	走行制御系	既製のソフトウェアを改良・改造	筑波大学知能ロボット研究室のyppspurを使用した。	Ubuntu 16.04で開発した。筑波大学知能ロボット研究室のSSMを使用し、プログラムのプロセス毎に分けて開発を行った。
			ドライバ	市販品	T-frog Projectの二輪ブラシレスモータドライバ(TF-2MD-R6)	センサ処理・認識系	自作システム PCL その他	センサデータをスキャンマッチングとマップ作成に使用した。スキャンマッチングの遅延最適化の手法に滑降シンプレックス法を用いた。可視化するために、PCLを用いた。	
1610	早稲田大学	天野研究室	走行部・ボディ	市販品を改造	株式会社メックデザインのNew Quattro 4Mをプラットフォームとして利用。レギュレーションを満たすように骨格の加工や部材の変更を行った。	金体システム	ROS	センサ系のドライバ等はオープンソースを利用し必要に応じて改造を行った。自己位置推定、環境認識、車両との通信部分に関してはノードとして自作した。ナビゲーションに関してはROSの標準パッケージに含まれているmove_baseを利用した。	ROS
		ARUMAJIRO	コントローラ	デスクトップPC	ShuttleのDS67U5を利用。省消費電力のPCでシリアルポートが利用可能なPCを選定。	走行制御系	ROS	ROSのmove_base利用により指令値(速度、角速度)の取得までは簡単にできたが、実際の車両制御のためには得られた指令値を実現させるようなモータの回転量に変換させる必要があるため、回帰式により車両モデルを作成した。	基本的にROSの機能を用いて開発を行った。特にbagファイルによるロボット内メッセージの記録・再生機能を多く利用した。各種パラメータの設定やアルゴリズムの見直しを使用した。
			ドライバ	利用した走行部に付随	PG VR2+RRC	センサ処理・認識系	Open-CV ROS	GNSS、レーザースキャナ、FOQ、エンコーダはROSにより処理を行った。有名なセンサであればドライバがパッケージとしてgitに存在していることがあるので、積極的に利用した。全天球カメラによる環境認識にはOpen-CVを利用しながら開発を進めているが、またロボットには実装できていない。	
1611	金沢プロジェクト	Adventure号	走行部・ボディ			金体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1612	明治大学	MORIOKA LAB.	走行部・ボディ	市販品を改造	Pioneer3-DX(MobileRobots/リバスト)をベースにした。	金体システム	ROS	ナビゲーションに関する既存パッケージの利用および人物探索のソフトウェアとの統合のため。	ROS
		オックスカル	コントローラ	ノートPC	東芝dynabook R73/PB	走行制御系	ROS	move_baseなどの既存走行制御ソフトを使用。	
			ドライバ	利用した走行部に付随	-	センサ処理・認識系	Open-CV ROS	主にCamelなどの自己位置推定ソフトを使用、一部は自作ソフトウェアも追加している。人物検出に関しては、OpenCVを使用。	
1613-01	小山高専	弓削商船高専	走行部・ボディ			金体システム			
		CompactBot	コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1613-02	小山高専	弓削商船高専	走行部・ボディ			金体システム			
		COYOMI	コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1614	大阪工業大学	情報科学部チーム	走行部・ボディ	市販品を改造	Kobuki(YujinRobot)をもとにしたTurtlebot2を改造して使用。	金体システム	ROS	-	ROS
		シリウス	コントローラ	デスクトップPC	GIGABYTE GB-BX5-4570R	走行制御系	ROS	-	
			ドライバ	利用した走行部に付随	-	センサ処理・認識系	ROS	-	
1615	横浜国立大学	藤本研究室	走行部・ボディ	市販品を改造	関東自動車工業 KMC-0001	金体システム	自作	-	自作
		PeGASuS	コントローラ	デスクトップPC	-	走行制御系	自作	-	
			ドライバ	自作	-	センサ処理・認識系	自作システム	-	














つくばチャレンジ2016 参加チーム一覧

No.	チーム名		1.ハードウェア			2.ソフトウェア		3.環境構築・シミュレータなど
	ロボット外観	ロボット内観など	走行部・ボディ	制御部	センサ処理・認識系	OS	ソフトウェア	
1616	芝浦工業大学 ロボティクス研究室	AND-2016	市販品を改造	i-Cart-mini	全体システム	ROS	2D : icart-mini-navigation, hokuyo_node, amcl, move_base 3D : hokuyo3d, octomap, pcl	ROS
			コントロール	ノートPC	Let's note CF-MX4	走行制御系	ROS	amcl,move_baseを使用。
			ドライブ	利用した走行部に付随	T-Frog TF-2MD3-R6	センサ処理・認識系	PCL ROS	hokuyo_node,hokuyo3d,octomapを使用。
1617	関西大学 ロボット・マイクロシステム研究室	KUARO	市販品を改造	MECROBOT 001 関大特別仕様	全体システム	ROS	全体のシステム統合にはROSを使用。主にROSの既存のパッケージを用いたが、一部は改良、自作した。	ROS
			コントロール	デスクトップPC	小型PC x 2 (Intel NUC5i7RYH)	走行制御系	自作 ROS	走行部に指令を与えて制御する部分は自作した。その他にはROSを使用し、地図作成にはgmapping、自己位置推定にはamcl、軌道計画にはmove_baseを使用した。
			ドライブ	利用した走行部に付随	オカテック社 TinyPower	センサ処理・認識系	自作システム OpenCV PCL ROS	センサのデータ処理にはOpen-CVやPCLを利用した。障害物や人物の認識は自作のプログラムを使用した。
1618	日本工業大学 石川研究室	Strayder	市販品を改造		全体システム			
			コントロール			走行制御系		
			ドライブ			センサ処理・認識系		
1619	チームさくら	強引My Way 2	市販品を改造	MERCEDES-BENZ SLK 製 子供乗用RC電動車 TOY-R81200	全体システム	自作 ROS	Frontier NXにてCroscoreを実行。地図作成時にLRFによりHectorSlamとRosBagを利用して地図作成。Raspberry Pi2にてLEDリングに点灯パターン変更。各種センサ情報を出力。自作走行時は、Let'sNoteにてエンコーダ情報を元に地図を基準として自己位置推定を行う。そしてLet's NoteからRaspberry Pi2にロボット制御データを送信し、PSoCにてモータ制御を行う。	自作 ROS
			コントロール	ノートPC シングルボードPC	Frontier NX Let's Note RaspberryPi2 Model B PSoC	走行制御系	自作	-
			ドライブ	市販品 利用した走行部に付随	スピードコントローラー HOBBYWING QuicRun-1060-Brushed サーボモーター JR PROPO NXB8921	センサ処理・認識系	自作システム ROS	地図作成時にLRFのセンサ情報を元にHectorSLAMを利用。
1620	長岡工業高校 ロボット部OB	エンデバー星外仕様2016 FINAL	市販品を改造		全体システム			
			コントロール			走行制御系		
			ドライブ			センサ処理・認識系		
1621	ALPS	ノリクラ	市販品を改造		全体システム			
			コントロール			走行制御系		
			ドライブ			センサ処理・認識系		
1622	WMMC つくばチャレンジプロジェクト	Capybara A.T.	市販品を改造		全体システム			
			コントロール			走行制御系		
			ドライブ			センサ処理・認識系		
1623-01	千葉大学 知能機械システム研究室	Cranberry 2016	自作	-	全体システム	自作	-	既製のソフトウェアを改良・改造
			コントロール	その他	-	走行制御系	自作	-
			ドライブ	市販品	オカテック社製 TinyPower	センサ処理・認識系	自作システム	-
1623-02	千葉大学 知能機械システム研究室	暁月拾陸	自作	-	全体システム	自作	-	既製のソフトウェアを改良・改造
			コントロール	その他	-	走行制御系	自作	-
			ドライブ	市販品	TinyPower(オカテック社製)	センサ処理・認識系	自作システム	-

つくばチャレンジ2016 参加チーム一覧

No.	チーム名		1.ハードウェア		2.ソフトウェア		3.環境構築・シミュレータなど
	ロボット外観	ロボット内観など					
1624	自律走行車プロジェクト(金沢工業大学)	コーボくんver2.1	走行部・ボディ		金体システム		
			コントローラ		走行制御系		
			ドライバ		センサ処理・認識系		
1625	Project C.G.S. Casybara J. V.		走行部・ボディ	市販品を改造	電動車いすズキMC3000Siに、外装(ボディ)や各種センサ・制御機器のマウントを自作。	金体システム	自作 ROS
			コントローラ	デスクトップ PC その他	マイコンボード(STM32F3 Discovery) : 電動車いすの制御 PC自作 : Core i7 3770S, DDR3 8GB, SSD 128GB) ; リアルタイム制御・センサなどデバイス管理 PC(市販品 : Intel NUC DC3Z17TY) ; 自己位置推定/プランニング/認識など上位層	走行制御系	自作
			ドライバ	利用した走行部に付随	電動車いすズキMC3000Sの制御系を流用。車いす従来のジョイスティックコントローラを外から制御値を入力できるように改造。	センサ処理・認識系	Open-CV ROS 本番走行で使用した部分は自作。ただし、未完信号物検出に関する部分はOpenCV+ROSを使用予定であった。
1626-01	群馬大学リバスターチーム	Mercury(Laser Model)	走行部・ボディ			金体システム	
			コントローラ			走行制御系	
			ドライバ			センサ処理・認識系	
1626-02	群馬大学リバスターチーム	Mercury(VisMo)	走行部・ボディ			金体システム	
			コントローラ			走行制御系	
			ドライバ			センサ処理・認識系	
1627	群馬大学・ミツパチーム	MG16	走行部・ボディ	自作	つくばチャレンジ2015に出場した際の車両 (MG15)	金体システム	自作
			コントローラ	ノートPC デスクトップ PC	MouseComputer 製のノートPC「LB-S222XSSD」; 車両制御用 Shuttle製の小型デスクトップ「XH61V」; 画像処理用	走行制御系	自作
			ドライバ	自作	Renesas製のSH2マイコンを搭載	センサ処理・認識系	Open-CV OS-基本ソフトウェア : Windows7 開発環境 : VisualStudio2008およびVisualStudio2010 ベースにしたシステム : つくばチャレンジ2015に出場した際の車両 (MG15) 開発環境 : VisualStudio2010 説明 : 走行実験で取得した撮影画像を用いて、参照画像に対する画像ナビゲーションのシミュレーションが可能
1628-01	尾崎研究室(宇都宮大学)	MAUV	走行部・ボディ	自作	-	金体システム	自作
			コントローラ	デスクトップ PC	-	走行制御系	自作
			ドライバ	利用した走行部に付随	-	センサ処理・認識系	自作システム Open-CV その他 信号認識や人探索といった画像処理のために、OpenCVやChainerを使っています。
1628-02	尾崎研究室(宇都宮大学)	SARA	走行部・ボディ	自作	-	金体システム	自作
			コントローラ	デスクトップ PC	-	走行制御系	自作
			ドライバ	利用した走行部に付随	-	センサ処理・認識系	自作システム GCCを用いてコンパイルしています。
1629-01	法政大学 自律移動ロボット実験室(ARL)	Orange2016	走行部・ボディ			金体システム	
			コントローラ			走行制御系	
			ドライバ			センサ処理・認識系	
1629-02	法政大学 自律移動ロボット実験室(ARL)	Orangemini2016	走行部・ボディ			金体システム	
			コントローラ			走行制御系	
			ドライバ			センサ処理・認識系	

つくばチャレンジ2016 参加チーム一覧













No.	チーム名		ロボットのプラットフォーム					
	ロボット外観	ロボット内観など	1.ハードウェア		2.ソフトウェア		3.環境関連・シミュレータなど	
1630	福岡大学フューチャージョン ロードランナー11型		走行部・ボディ	自作	走行部として、GoldenMotor社製電動自転車用車輪ユニットおよびコントローラを使用。	全体システム	自作	
			コントローラ	ノートPC	-	走行制御系	自作	
			ドライバ	利用した走行部に付随	-	センサ処理・認識系	自作システム	
1631	筑波大学知能ロボット研究室 チームRossy Rossy		走行部・ボディ	自作	-	全体システム	ROS	ROS Kinsito Kameを使用。
			コントローラ	デスクトップPC	ALINEWARE (DELL) 製 X51 R3	走行制御系	既製のソフトウェアを改良・改造 ROS	経路計画 ROS move_base、モータコントローラ YP-Spur
			ドライバ	市販品	T-frogProject TF-2MD3-R6	センサ処理・認識系	Open-CV PCL ROS その他	
1632	筑波大学知能ロボット研究室 チームKerberos Kerberos		走行部・ボディ	市販品を改造	昨年のつくばチャレンジ2015に参加した同チームのロボットをベースとして改造した。	全体システム	ROS	
			コントローラ	デスクトップPC	Shuttle XH170V (Core-i7 6700)	走行制御系	ROS	ROSのmove_baseパッケージを利用した。
			ドライバ	市販品	二輪モータドライバ：TF-2MD3-R6	センサ処理・認識系	ROS	障害物回避のためにvelodyne_height_mapパッケージを使用。探索対象認識のための学習を行なうために、NVIDIA DIGITSを利用した。
1633	筑波大学知能ロボット研究室 チームKenaf Kenaf		走行部・ボディ			全体システム		
			コントローラ			走行制御系		
			ドライバ			センサ処理・認識系		
1634	ロボットタスクシステム研究室 (芝浦工業大学) ρ-キューブ		走行部・ボディ	市販品を改造	i-Cart mini	全体システム	ROS	「使えるものは何でも使う」「車輪の再開発は避ける」をモットーにした。その結果、今年度のシステムは既存のROSパッケージのみでの構成となった。
			コントローラ	ノートPC	Panasonic Let's note CF-SX4KFTBR	走行制御系	ROS	move_base, amcl, ypspur_ros, bridgeを使った。
			ドライバ	利用した走行部に付随	TF-2MD3-R6	センサ処理・認識系	ROS	Top-URGのドライバにはurg_nodeを使用した。環境地図の作成にはslam_gmappingを使用した。
1635	神奈川工科大学 吉留研究室 KAIT3世		走行部・ボディ	自作	-	全体システム	自作	
			コントローラ	デスクトップPC	Epson ST180E	走行制御系	自作	
			ドライバ	市販品	Pololu RoboClaw 2x60A	センサ処理・認識系	自作システム	
1636	防衛大学校 滝田・富沢研究室 AR Skipper		走行部・ボディ	自作	-	全体システム	自作	自作
			コントローラ	ノートPC	-	走行制御系	自作	
			ドライバ	自作	-	センサ処理・認識系	自作システム	
1637	HORIO JAPAN HORIO KART		走行部・ボディ			全体システム		
			コントローラ			走行制御系		
			ドライバ			センサ処理・認識系		
1638	KIRT ALBERT		走行部・ボディ			全体システム		
			コントローラ			走行制御系		
			ドライバ			センサ処理・認識系		



つくばチャレンジ2016 参加チーム一覧

No.	チーム名		ロボットのプラットフォーム						
	ロボット外観	ロボット内観など	1.ハードウェア			2.ソフトウェア		3.駆動両輪・シミュレータなど	
1639	つくばほっとサークル TRC-20		走行部・ボディ	自作	2013年度参加時の自サークルの機体。	全体システム	ROS	ROS Kinetic Kameを使用。 Arduinoとの通信にrosserialを使用。	ROS
			コントローラ	ノートPC	Lenovo E440	走行制御系	自作 ROS	モータ制御自体は自作、 速度の指定にROSを使用した。	開発のベースはROSを使用。 シミュレータにはrvizを使用。
			ドライバ	自作	-	センサ処理・認識系	ROS	北陽電気Top-URGを使用、その際hokuyo_nodeを使用。	
1640	VACCOS 出掛け太郎 番号機		走行部・ボディ	市販品を改造	メーカー：ハイガー産業 製品名：BEST LIFE(ベストライフ) 型式：HG-DWAC01S	全体システム	既製のソフトウェアを改良・改造	Microsoft Windows 10：各種認識処理、全体制御 Microsoft Windows 10 IoT：車体制御用Raspberry Pi用OS Microsoft Windows 10 Mobile：スマートフォンをセンサ群として使用	既製のソフトウェアを改良・改造
			コントローラ	シングルボードPC	Raspberry Pi 3	走行制御系	既製のソフトウェアを改良・改造	進行方向や前進後退を決定するPCからの制御バケットを受信したRaspberry Piが、接続されたサーボモーターを動かすことで、車両本体のアクセル(可変抵抗器)やハンドル(メカニカル)を物理的に操作する制御方式。	Visual Studio 2013：画像処理用Win32アプリ作成用、スマートファンアプリ作成用 Visual Studio 2015：UWPアプリ作成用(Raspberry Pi制御用、通信用)
			ドライバ	自作	-	センサ処理・認識系	自作システム Open-CV	進行方向検出、物体検出のための画像処理：自作+一部でOpenCVを使用 前方障害物検出用：Kinect for Windows v2 センサ情報収集・処理：スマートフォン内蔵のほぼ全てのセンサを使用。 データ収集・処理・通信は自作のドライブレコーダアプリを改良。	
1641	NU Intelligent Mobility Lab. CARTIS		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1642	AMSL Racing (明治大) INFANT		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1643	明星大学 飯島研・山崎研 Bright Star 2016		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1644	千葉工業大学 fuRo アウトドア部Ⅱ Mair 2		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1645	金沢高専 ItolabHouse		走行部・ボディ	自作	月面ローバ	全体システム	自作	-	既製のソフトウェアを改良・改造
			コントローラ	シングルボードPC	Nvidia Jetson TK1	走行制御系	自作	比例航法を用いて目的地まで走行。	車体にモーションセンサ(東京航空計器：CSM-MG100)、ウェアラブルカメラ(Panasonic-HX-A1H)、測域センサ(北陽電機UTM-30LX)を搭載し、センサとして使用。 コンピュータにはNvidia Jetson TK1を用いて、Ubuntu14.04LTS、OpenCV3.00、PCL1.7.0:library1.2をインストールして使用している。
			ドライバ	自作	-	センサ処理・認識系	既製のソフトウェアを改良・改造	測域センサ(北陽電機UTM-30LX)を利用し、障害物回避を行う。また、OpenCVを用いて画像処理を行い三次元マップを作成する。	
1646	千葉工業大学 藤田研 SFLABOT		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1647	速える子羊大学 メリー教授		走行部・ボディ	自作	モーターはT-Frogプロジェクト製"TF-M30-24-3500-G15"を使用。	全体システム	自作 ROS	デバイスハンドラはROS標準pkgをそのまま利用。 またROS標準のナビゲーションの枠組み(move_base)を利用したが、実装部分は自作。	自作 ROS
			コントローラ	ノートPC	-	走行制御系	既製のソフトウェアを改良・改造	yaspurコマンドを利用(cmd_vel, get_odom, get_AD)、オドメトリの算出、経路追従制御系を自作している。	キネマティクスレベルのシミュレータを自作。 ソフトウェア開発環境は、 ・Ubuntu 14.04LTS ・Eclipse CDT + PyDev
			ドライバ	市販品	モータードライバはT-Frogプロジェクト製"TF-2MD3-R8"を使用。 ギアボックスは30:1仕様へ換装。	センサ処理・認識系	Open-CV PCL	フィルタリングにはrosのlaser_filters枠組みを利用し、プラグイン形式の実装部を自作。また"地面境界線抽出ロジック"などは上記で挙げた各種OSSを利用。	

つくばチャレンジ2016 参加チーム一覧

No.	チーム名		ロボットのプラットフォーム			3. 環境開発・シミュレータなど			
	ロボット外観	ロボット内観など	1. ハードウェア		2. ソフトウェア				
1648	大阪大学 みらいロボットユニオンS A.V.A.Y.O.		走行部・ボディ	市販品を改造	ベース台車 リバース製 Mercury Mega Tsukuba Model	全体システム	自作	-	自作
			コントローラ	ノートPC	HP Z-BOOK15 G2 をメインとし、それとほぼ同等性能のラップトップPCをサブとして用いた。	走行制御系	自作	-	各プロセスにファイルからデータをリプレイする機能を付加し、実験時と同様の処理をオンラインでシミュレート可能なシステムを構築した。
			ドライバ	利用した走行部に付随	-	センサ処理・認識系	Open-CV POL	カメラからの画像処理にOpen CV、3D LRFのデータ処理にPOLを用いた。	
1649	愛知工業大学 道木研究室 AIT-energy® Monster		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1650-01	中後研・村松研・横田研合同チーム (関西学院大・東海大・東洋大) 走ルンでず1号		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1650-02	中後研・村松研・横田研合同チーム (関西学院大・東海大・東洋大) UGV16		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1651	東京高専ロボティクス連携チーム 高尾5号		走行部・ボディ	自作		全体システム	ROS	自己位置推定: amcl 地図作製: slam_gmapping ナビゲーション: move_base ビジュアライゼーション: rviz LRF統合: ira_laser_tools	ROS
			コントローラ	デスクトップPC	BOXNUCSI7RYH (Intel)	走行制御系	ROS	自己位置推定: amcl 地図作製: slam_gmapping ナビゲーション: move_base	OS: ROS Indigo Iglou on Linux Ubuntu 14.04 64-bit 環境開発: ROS Indigo Iglou
			ドライバ	市販品	KENTAC3815 (昭和電業)	センサ処理・認識系	ROS	LRF統合: ira_laser_tools	
1652	CIR-KIT A KIT-C3		走行部・ボディ	市販品を改造	スズキ製セニアカー タウンカート (TC1A4)	全体システム	自作 ROS	ROSのパッケージで利用できるものは利用した。ただし、ロボット専用のドライバや人物探索機能などはROSのパッケージとして開発した。	ROS
			コントローラ	ノートPC その他	iXis research社製 iMCs01 Arduino Uno	走行制御系	自作 ROS	ROSのnavigationスタックを利用した。Gmapping, amcl, move_baseを利用した。ただし、move_baseの中のローカルプランナはdwa_local_plannerに変更し、グローバルプランナはcarrot_plannerに変更した。waypointの作成やwaypointの管理には自作のソフトウェアを開発した。	全体のシステムはUbuntu14.04上のROS(indigo)上で開発した。また、gazeboを利用してシミュレーション環境を構築した。
			ドライバ	利用した走行部に付随	セニアカーに搭載されていたものを用いている。	センサ処理・認識系	PCL ROS その他	2D、3D LIDARのドライバはROSで提供されているものを利用した。また、3D LIDARで得られた点群を処理するためにPOLで提供されている関数を利用した。人物探索にSVMを用いたが実装にはLbSVMを利用した。	
1653-01	CIR-KIT B KIT-C4		走行部・ボディ			全体システム			
			コントローラ			走行制御系			
			ドライバ			センサ処理・認識系			
1653-02	CIR-KIT B KIT-C5		走行部・ボディ	市販品を改造	T-frog Project I-Cart mini	全体システム	ROS	ROSのNavigationを利用した。	ROS
			コントローラ	ノートPC	-	走行制御系	ROS	ROS NavigationからのメッセージをYo-Spurに送って制御した。	全体のシステムはUbuntu16.04上のROS(Kinetic)上で開発した。シミュレーションは使用しなかった。
			ドライバ	市販品	二軸モータードライバ TF-2MD3-R6	センサ処理・認識系	ROS	2D LRFのドライバはROSで提供されているものを利用した。	