

各種APIリファレンス

- ・ Create用API一覧
- ・ SD用API一覧
- ・ 超音波距離センサ用API一覧
- ・ 参考資料

Create センサ系API

API	説明	引数	戻り値
boolean Create::isRightBumperPushed(void)	右バンパーの状態を取得	なし	true: 非接触状態 False: 接触状態
boolean Create::isLeftBumperPushed(void)	左バンパーの状態を取得	なし	true: 非接触状態 False: 接触状態
uint8_t Create::getIRByte(void)	ドッグからの赤外線受信状態を取得 (参考資料1 参照)	なし	以下のいずれかの値を返却 Create::IRCommandRedBuoy Create::IRCommandGreenBuoy Create::IRCommandForceField Create::IRCommandRedGreenBuoy Create::IRCommandRedBuoyForceField Create::IRCommandGreenBuoyForceField Create::IRCommandRedGreenBuoyForceField
boolean Create::isPlayButtonPushed(void)	Playボタンの状態を取得	なし	0: 非押下状態 1: 押下状態
boolean Create::isAdvanceButtonPushed(void)	Advanceボタンの状態を取得	なし	0: 非押下状態 1: 押下状態
uint16_t Create::getVelocity(void)	前回drive()にて指定された速度を取得	なし	-500 - 500 mm/s
uint16_t Create::getTurningRadius(void)	前回drive()にて指定された回転半径を取得	なし	-32768 - 32767 mm
uint16_t Create::getDistance(void)	前回のgetDistance()実行時からの 前進距離を取得	なし	-32768 - 32767 mm 正の数: 前進 負の数: 後退
uint16_t Create::getAngle(void)	前回のgetAngle()実行時からの 回転角度を取得	なし	-32768 - 32767 mm 正の数: 逆時計回り 負の数: 時計回り
uint8_t Create::getChargingState(void)	現在の充電状態を取得	なし	以下のいずれかの値を返却 Create::ChargeStateNotCharging Create::ChargeStateReconditioningCharging1 Create::ChargeStateFullCharging Create::ChargeStateTrickleCharging Create::ChargeStateWaiting Create::ChargeStateFault
uint16_t Create::getCharge(void)	現在のバッテリー残量を取得	なし	0 - 65535 mAh
uint8_t Create::getWall(void)	壁検知状態を取得	なし	0: 壁非検知状態 1: 壁検知状態

Create スタート制御系API

API	説明	引数	戻り値
void Create::start(void)	passiveモードでCreateをスタート	なし	なし
void Create::startWithSafeMode(void)	safeモードでCreateをスタート	なし	なし
void Create::reset(void)	Createをリセット	なし	なし
void Create::safeMode(void)	Createをsafeモードに移行	なし	なし
void Create::fullMode(void)	Createをfullモードに移行	なし	なし

※ 動作モードに関しては参考資料2を参照

Create 動作制御系API

API	説明	引数	戻り値
void Create::drive(int16_t velocity, int16_t radius)	Createのタイヤを、半径radius[mm]の弧を 速度velocity[mm/s]で走行するよう回転させる	uint16_t velocity: 速度(-500 – 500 mm/s) uint16_t radius: 回転半径(-2000 – 2000 mm) (※)radiusには、半径の代わりに以下を指定することも可能 Create::DriveStraight : 直進 Create::DriveInPlaceClockwise : その場で時計回りに回転 Create::DriveInPlaceCounterClockwise : その場で反時計回りに回転	なし
void Create::driveDirect(int16_t leftVelocity, int16_t rightVelocity)	Createの左右のタイヤを指定した速度で 回転させる	int16_t leftVelocity: 左タイヤの回転速度(-500 – 500 mm/s) int16_t rightVelocity: 右タイヤの回転速度(-500 – 500 mm/s)	なし
void Create::wait(uint8_t ticks)	指定した時間だけCreateをwaitさせる	uint8_t ticks: waitさせる時間(tick, 1tick = 15ms)	なし
void Create::waitDistance(int16_t mm)	指定した距離を進むまでCreateをwaitさせる Createはwaitが完了するまで他のコマンドを 受け付けけない	uint16_t mm: waitする距離(mm)	なし
void Create::waitAngle(int16_t degrees)	指定した角度だけ回転するまでCreateを Waitさせる Createはwaitが完了するまで他のコマンドを 受け付けけない	uint16_t degrees: waitする角度(degrees)	なし
void Create::dock(void)	ドッグを探索し、ドッキングする	なし	なし

SDWriter API

API	説明	引数	戻り値
void SDWriterClass::init(char* filename)	ファイルを初期化する	filename: ファイルを指定	なし
int SDWriterClass::open(const char* filename, Uint8_t mode = FILE_WRITE)	ファイルをオープンする	filename: ファイルを指 mode: FILE_READ=読み込みでオープン FILE_WRITE=書き込みでオープン	正常: 0 失敗: -1
void SDWriterClass::close(void)	ファイルをクローズする	なし	なし
void SDWriterClass::print(const char* buf)	現在オープンされているファイルに文字列を書き込む	buf: 書き込む文字列を指定	なし
void SDWriterClass::printf(const char* format, ...)	現在オープンされているファイルにフォーマットで指定した文字列を書き込む	format: 書き込む文字列のフォーマットを指定 (#printfと同等のフォーマットが使用可能) ...: フォーマットで指定したとおりの変数を指定	なし
void SDWriterClass::printf(const char* filename, Const char* format, ...);	ファイルをオープンし、ファイルにフォーマットで指定した文字列を書き込む	filename: ファイルを指定 format: 書き込む文字列のフォーマットを指定 (#printfと同等のフォーマットが使用可能) ...: フォーマットで指定したとおりの変数を指定	なし

CreateDataSD API

API	説明	引数	戻り値
void CreateDataSDWriterClass::init(void)	data.csvの初期化を行う	なし	なし
void CreateDataSDWriterClass::printCreateData(Create create)	data.csvにCreateのデータを書き込む	createのオブジェクトを指定	なし

SDReader API

API	説明	引数	戻り値
int SDReaderClass::open(const char* filename, Uint8_t mode = FILE_READ)	ファイルをオープンする	filename: ファイルを指定 mode: FILE_READ=読み込みでオープン FILE_WRITE=書き込みでオープン	なし
void SDReaderClass::close(void)	ファイルをクローズする	なし	なし
void SDReaderClass::readln(char* str)	オープンしたファイルから1行読み出す	str: 読み込んだ文字列を格納するバッファを指定	なし

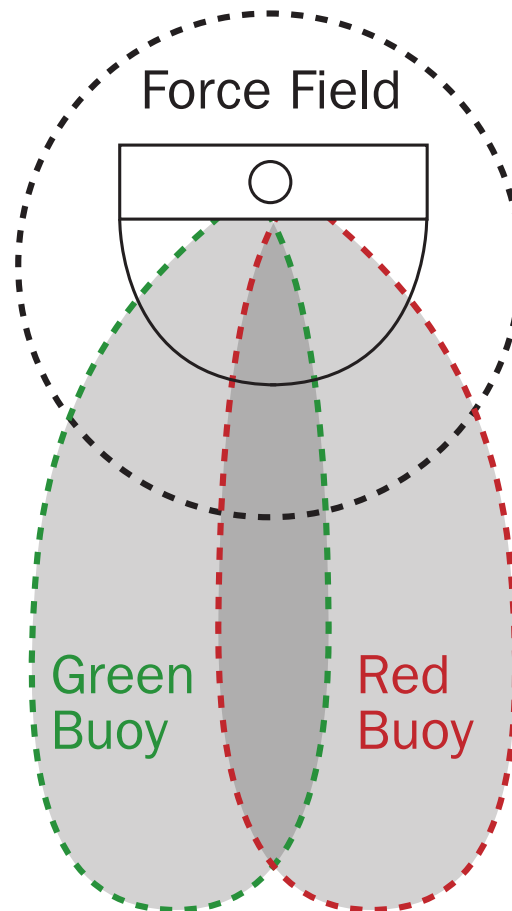
超音波距離センサ API

API	説明	引数	戻り値
int SonarRangeFinder::getRange(void)	対象物までの距離を取得する	なし	対象物までの距離(mm)

参考資料1

ドッグからのIR出力

Dock beam configuration



参考資料2 Createの動作モード

The Create OI has four operating modes: Off, Passive, Safe, and Full. After a battery change or when is first supplied, the OI is in “off” mode. When it is off, the OI listens at the default baud rate (57600 or 19200 - see Serial Port Settings above) for an OI Start command. Once it receives the Start command, you can enter into any one of the four operating modes by sending a mode command to the OI. You can also switch between operating modes at any time by sending a command to the OI for the operating mode that you want to use.

Passive Mode

Upon sending the Start command or any one of the demo commands (which also starts the specific demo, e.g., Spot Cover, Cover, Cover and Dock, or Demo), the OI enters into Passive mode. When the OI is in Passive mode, you can request and receive sensor data using any of the sensors commands, but you cannot change the current command parameters for the actuators (motors, speaker, lights, low side drivers, digital outputs) to something else. To change how one of the actuators operates, you must switch from Passive mode to Full mode or Safe mode.

While in Passive mode, you can read Create’s sensors, watch Create perform any one of its ten built-in demos, and charge the battery.

Safe Mode

When you send a Safe command to the OI, Create enters into Safe mode. Safe mode gives you full control of Create, with the exception of the following safety-related conditions:

- Detection of a cliff while moving forward (or moving backward with a small turning radius, less than one robot radius).
- Detection of a wheel drop (on any wheel).
- Charger plugged in and powered.

Should one of the above safety-related conditions occur while the OI is in Safe mode, Create stops all motors and reverts to the Passive mode.

If no commands are sent to the OI when in Safe mode, Create waits with all motors and LEDs off and does not respond to Play or Advance button presses or other sensor input.

Note that charging terminates when you enter Safe Mode.

Full Mode

When you send a Full command to the OI, Create enters into Full mode. Full mode gives you complete control over Create, all of its actuators, and all of the safety-related conditions that are restricted when the OI is in Safe mode, as Full mode shuts off the cliff, wheel-drop and internal charger safety features. To put the OI back into Safe mode, you must send the Safe command.

If no commands are sent to the OI when in Full mode, Create waits with all motors and LEDs off and does not respond to Play or Advance button presses or other sensor input.

Note that charging terminates when you enter Full Mode.