LED-Camp3 API リファレンス集 v2

LED-Camp 実行委員会

※資料集に掲載のものからの差分は，赤字で示している.

0. 目次

注：本資料では，LED-Camp3 実習に使用する API のみを説明しています．実装・提供している全ての API をカバーしているわけではありません.
文中の「仕様書」は，「iRobot® Roomba Open Interface (OI) Specification」を指します.

1. iRobot Create2 用 API

./arduino/lib/LEDCamp3API/iRobotCreate2.h

1.1. 初期化・スタート制御系

○ iRobotCreate2* iRobotCreate2::getInstance(void)

【動作・意味】iRobotCreate2 クラスのインスタンスを生成する（シングルトン用）
【引数】無し
【返値】iRobotCraete2*: iRobotCreate2 クラスのインスタンス
【使用例】

```
// create2 を宣言してインスタンスを取得する
iRobotCreate2* create2 = iRobotCreate2::getInstance();
```

○ void iRobotCreate2::start(void)

【動作・意味】iRobotCreate2 OI コマンドの送受信を開始する．Create2 用 API の実行前に必ず呼び出す必要がある．

さらに，各種センサの取得値を初期化した後に，センサの値を周期的に取得する動作を開始する．（動作の周期は 50ms）

【引数・返値】無し

○ void iRobotCreate2::safeMode(void)

【動作・意味】Create2 を safe モードに移行させる．

【引数・返値】無し

○ void iRobotCreate2::fullMode(void)

【動作・意味】Create2 を full モードに移行させる．

【引数・返値】無し

○ void iRobotCreate2::passiveMode(void)

【動作・意味】Create2 を safe モードに移行させる．

【引数・返値】無し

○ void iRobotCreate2::stop(void)

【動作・意味】iRobotCreate2 OI コマンドの送受信を終了する．

【引数・返値】無し

○ void iRobotCreate2::reset(void)

【意味】Create2 の電源を OFF にする．

【引数・返値】無し

○ void iRobotCreate2::turnOff(void)

【意味】Create2 のデバイスをリセットする．バッテリ交換時のみに実行する必要がある．

【引数・返値】無し

1.2.　モータ駆動系

○ void iRobotCreate2::drive(int velocity, int radius)

【動作・意味】Create2 を指定の回転速度および回転角度で動作させる.

【引数】int velocity: 回転速度. 単位は[mm/s]（-500～500 で指定可能）

int radius: 回転半径. 単位は[mm]（-2000～2000 で指定可能）

正の値が反時計回りとなる

【返値】無し

○ void iRobotCreate2::driveWheels(int right, int left)

【動作・意味】Create2 の両輪を指定の回転速度で動作させる.

【引数】int right: 右車輪の回転速度. 単位は[mm/s]（-500～500 で指定可能）

int left: 左車輪の回転速度. 単位は[mm/s]（-500～500 で指定可能）

【返値】無し

○ void iRobotCreate2::driveWheelsPWM(int rightPWM, int leftPWM)

【動作・意味】Create2 の両輪を指定の PWM デューティ値で動作させる.

【引数】int right: 右車輪の PWM デューティ値.（-255～255 で指定可能）

int left: 左車輪の PWM デューティ値.（-255～255 で指定可能）

【返値】無し

1.3.　本体センサ系

○ void iRobotCreate2::setSensorListener(
void *(listener)(unsigned long), int funcNumber)

【動作・意味】センサ値に変化があった場合に呼び出される関数を登録する.

【引数】void *(listener)(unsigned long): 登録する関数

void listener(unsigned long)型の関数を登録できる.

int funcNumber: 関数の登録先の番号（0～9 まで指定可能）

【返値】無し

○ long iRobotCreate2::getDistance(void)

【動作・意味】現在の移動距離を取得する.

【引数】無し

【返値】long: 現在の移動距離. 単位は mm.

○ long iRobotCreate2::getAngle(void)

【動作・意味】現在の角度を取得する.

【引数】無し

【返値】long: 現在の角度. 単位は度.

正の値は反時計回りの方向となる.

○ long iRobotCreate2::getRightEncoder(void)

【動作・意味】Create2 の右車輪のエンコーダのカウント値を取得する.

【引数】無し

【返値】long: 右車輪のエンコーダのカウント値.

1 周辺り 508.8 カウント(タイヤの直径は 72mm なので 1 カウント辺り 0.444mm 回転)

○ long iRobotCreate2::getLeftEncoder(void)

【動作・意味】Create2 の左車輪のエンコーダのカウント値を取得する.

【引数】無し

【返値】long: 左車輪のエンコーダのカウント値. 同上.

○ void iRobotCreate2::setNextDistance(long distance)

【動作・意味】指定した距離を超えて移動した際にイベント ReachDistance を発生させる.

【引数】long distance: 距離の指定. 単位は mm.

【返値】無し

○ void iRobotCreate2::setNextAngle(long angle)

【動作・意味】指定した角度を超えて移動した際にイベント ReachAngle を発生させる.

【引数】long distance: 距離の指定. 単位は度.

正の値は反時計回りの方向となる.

【返値】無し

1.4. 本体ボタン・音楽・LED 表示系

○ void iRobotCreate2::pushButton(byte button)

【動作・意味】引数に該当する create2 本体のボタンの押下動作を実行する.

【引数】byte button: 該当ボタン (8 ビットの 2 進数表現)

本体ボタン (8 個) とビットの対応は, 仕様書の Page 16 を参照.

【返値】無し

○ void iRobotCreate2::createSong(byte songNumber,
byte numberOfNotes, Notes notes[])

【動作・意味】Create2 に音楽を登録する.

【引数】byte songNumber: 音楽の登録先の番号（0～4 まで指定可能）

byte numberOfNotes: 音楽の長さ（1～16 まで指定可能）

Notes notes[]: 登録する音楽の音符の配列

notes[].note: ピッチ, 音の高さ（31～127 まで指定可能）

notes[].duration: デュレーション, 音の長さ（0～255 まで指定可能）

【返値】無し

○ void iRobotCreate2::playSong(byte songNumber)

【動作・意味】Create2 に登録した音楽を再生する.

【引数】byte songNumber: 登録した音楽の番号

【返値】無し

○ void iRobotCreate2::setLEDs(byte ledBits, byte color, byte intensity)

【動作・意味】Create2 本体の LED を点灯させる.

Home LED（中央）は色と輝度を指定する. その他の LED は ON/OFF を指定する.

【引数】byte ledBits: 該当するその他の LED（4 ビットの 2 進数表現）

本体 LED（4 個）とビットの対応は, 仕様書の Page 15 を参照.

byte color: Home LED の色（0～255 まで指定可能. 0 は緑, 255 は赤）

byte intensity: Home LED の輝度（0-255 まで）

【返値】無し

○ void iRobotCreate2::setDigitLEDs(byte digit3, byte digit2,
byte digit1, byte digit0)

【動作・意味】Create2 本体の 7 セグ LED（4 桁）を点灯させる.

【引数】byte digit3: 3 桁目の 7 セグ LED（7 ビットの 2 進数表現）

byte digit2: 2 桁目の 7 セグ LED（7 ビットの 2 進数表現）

byte digit1: 1 桁目の 7 セグ LED（7 ビットの 2 進数表現）

byte digit0: 0 桁目の 7 セグ LED（7 ビットの 2 進数表現）

7 セグ LED の各セグメント（7 個）とビットの対応は, 仕様書の Page 16 を参照.

【返値】無し

○ void iRobotCreate2::setDigitLEDFromASCII (byte digit3, byte digit2, byte digit1, byte digit0)

【動作・意味】Create2 本体の 7 セグ LED（4 桁）を指定された ASCII コードで点灯させる.

【引数】byte digit3: 3 桁目の 7 セグ LED の ASCII コード（32～126 まで指定可能）

byte digit2: 2 桁目の 7 セグ LED の ASCII コード（32～126 まで指定可能）

byte digit1: 1 桁目の 7 セグ LED の ASCII コード（32～126 まで指定可能）

byte digit0: 0 桁目の 7 セグ LED の ASCII コード（32～126 まで指定可能）

7 セグ LED と ASCII コードの対応は，仕様書の Page 17 を参照.

【返値】無し

2. GR-SAKURA 本体用用 API（Arduino 互換 API）

./arduino/core/Arduino.h

2.1. デジタル入出力系

○ void pinMode(uint8_t pin, uint8_t mode)

【動作・意味】デジタルピンの動作を入力か出力かに設定する.

【引数】uint8_t pin: 設定対象のピン番号

デジタルピンのコネクタ：PIN_IO0～PIN_IO13 まで

GR-SAKURA 上の LED：PIN_LED0～PIN_LED3

GR-SAKURA 上のプッシュスイッチ：PIN_SW

uint8_t mode: 設定する動作. INPUT: 入力, OUTPUT: 出力

【返値】無し

○ void digitalWrite(uint8_t pin, uint8_t value)

【動作・意味】指定したピンに HIGH または LOW を出力する.

【引数】uint8_t pin: 出力に指定するピン番号（0～13 まで指定可能）

uint8_t value: 出力値. HIGH: 1, LOW: 0

【返値】無し

○ int digitalRead(uint8_t pin)

【動作・意味】指定したピンの値を読み取る.

【引数】uint8_t pin: 読み取り対象のピン番号（0～13 まで指定可能）

【返値】int: 指定ピンの値（HIGH: 1 または LOW: 0）

2.2.　時間系

○ void delay(unsigned long ms)

【動作・意味】指定時間だけプログラムの動作をミリ秒単位で停止する.

【引数】unsigned long ms: 一時停止する時間. 単位はミリ秒.

【返値】無し

○ void delayMicroseconds(unsigned long us)

【動作・意味】指定時間だけプログラムの動作をマイクロ秒単位で停止する.

【引数】unsigned long us: 一時停止する時間. 単位はマイクロ秒.

【返値】無し

2.3.　シリアル通信系

./arduino/core/HardwareSerial.h

○ void Serial.begin(unsigned long baud)

【動作・意味】シリアル通信のデータ転送レートを bps で指定する.

【引数】unsigned long baud: データ転送レート

【返値】無し

○ void Serial.print(data, int format)

【動作・意味】ASCII テキストのデータをシリアルポートに出力する.

【引数】data: 出力データ

int format: データ表示の基数（省略可能）

BIN: 2 進数　OCT: 8 進数　DEC: 10 進数　HEX: 16 進数

【返値】無し

○ void Serial.println(String data, int format)

【動作・意味】テキストメッセージを文末に改行付きでシリアルポートに出力する.

【引数】String data: 出力メッセージ（整数型も可能）

int format: データ表示の基数（省略可能）

BIN: 2 進数　OCT: 8 進数　DEC: 10 進数　HEX: 16 進数

【返値】無し

3. 超音波センサ用 API

./arduino/lib/LEDCamp3API/SonicSensor.h

○ SonicSensor SonicSensor(int trig, int echo, float coefficient)

【動作・意味】超音波センサクラスのインスタンスを生成する.

【引数】int trig: トリガ用に使用するピンの指定

int echo: エコー用に使用するピンの指定

float coefficient: 取得されるセンサ値の補正係数（省略可能）

省略時はデフォルト値の 1.0924 が設定される

【返値】超音波センサクラス SonicSensor のインスタンス

【使用例】

```
// sonicsensor のインスタンスを宣言する
SonicSensor sonicsensor(9,8);
```

○ float SonicSensor::getSonicSensor(void)

【動作・意味】超音波センサの値を取得する.

【引数】無し

【返値】float: 超音波センサの値. 単位は cm.

センサ性能の制約上, 3.0cm 以内または 100.0cm 以上を検知した場合は-1 を返す.

○ void SonicSensor::setCoefficient(float coefficient)

【動作・意味】超音波センサの補正係数を変更する.

【引数】float coefficient: 取得されるセンサ値の補正係数

【返値】無し

○ void iRobotCreate2::setNextSonicDistance(float sonicdistance)

【動作・意味】超音波センサの値が指定した値未満となった際に

イベント ReachSonicDistance を発生させる.

【引数】float sonicdistance: 超音波センサの値の指定

【返値】無し

【補足】他の超音波センサ用 API とクラスが異なることに注意すること.

4. タイマ用 API

./arduino/lib/LEDCamp3API/Timer.h

○ Timer* Timer::getInstance(void)

【動作・意味】Timer クラスのインスタンスを生成する（シングルトン用）

【引数】無し

【返値】Timer*: Timer クラスのインスタンス

【使用例】

```
// timer を宣言してインスタンスを取得する
Timer* timer = Timer::getInstance();
```

○ unsigned long timer::getTime(void)

【動作・意味】インスタンスの生成時またはリセット時からの経過時間をミリ秒で取得する.

【引数】無し

【返値】unsigned long: 経過時間. 単位はミリ秒.

○ void timer::resetTime(void)

【動作・意味】タイマの経過時間をリセットする.

【引数】無し

【返値】無し

○ void Timer::setNextTimeout(unsigned long timeout)

【動作・意味】指定した時間が経過した際にイベント Timeout を発生させる.

【引数】unsigned long:経過時間の指定. 単位はミリ秒

【返値】無し

【補足】イベント Timeout は, MDD プラグインから生成される Events.h では
Controller_Timeout の名前にリネームされる.