

# LED-Camp6 API リファレンス集

注：本資料では、LED-Camp の実習に使用する Controller クラスの API を中心に説明しています。ライブラリにはその他の API も実装されていますが、直接実習では使用しません。

## 1. Controller API

### 1.1 getPosition

【関数名】	getPosition
【動作・意味】	LEDTank の移動距離、旋回角度を取得します。
【引数】	[out]float *distance: 移動距離[cm] [out]float *angle: 旋回角度[deg] (時計周りが+, 反時計周りが-)
【返値】	無し

### 1.2 reset

【関数名】	reset
【動作・意味】	LEDTank の移動距離、旋回角度の累積をリセットします。
【引数】	無し
【返値】	無し

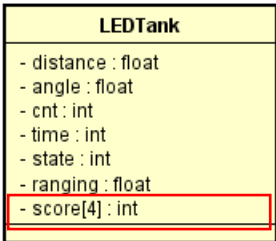
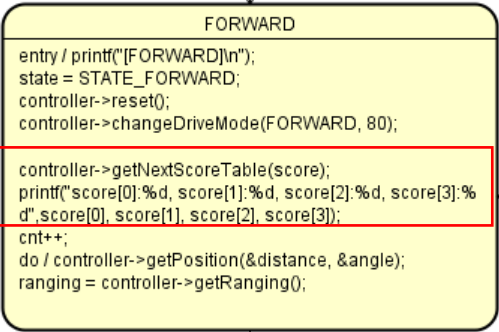
### 1.3 changeDriveMode

【関数名】	changeDriveMode
【動作・意味】	モータの駆動モードと出力を変更します。
【引数】	[in]Mode mode: 駆動モード ・STOP: 停止 ・FORWARD: 前進 ・BACK: 後退 ・CW: 時計周りに旋回 ・CCW: 反時計回りに旋回 [in]int voltage_level: 電圧レベル (弱 0~100 強を指定)
【返値】	無し

## 1.4 getRanging

【関数名】	getRanging
【動作・意味】	Ranging センサーの値を取得します。
【引数】	無し
【返値】	float Rangin センサー値[cm]

## 1.5 getNextScoreTable

【関数名】	getNextScoreTable
【動作・意味】	<p>スコアテーブルを取得します。 点数は配列に格納されます。 nextScoreTable[0]: 目標① nextScoreTable[1]: 目標② nextScoreTable[2]: 目標③ nextScoreTable[3]: 目標④</p> <p>※クラス図の変数の追加例</p>  <pre> classDiagram     class LEDTank {         - distance : float         - angle : float         - cnt : int         - time : int         - state : int         - ranging : float         - score[4] : int     }     </pre> <p>※ステートマシン図の中での使用例</p>  <pre> stateDiagram-v2     [*] --&gt; FORWARD     FORWARD: entry / printf("[FORWARD]\n");\nstate = STATE_FORWARD;\ncontroller-&gt;reset();\ncontroller-&gt;changeDriveMode(FORWARD, 80);\ncontroller-&gt;getNextScoreTable(score);\nprintf("score[0]:%d, score[1]:%d, score[2]:%d, score[3]:%d", score[0], score[1], score[2], score[3]);\ncnt++;\ndo / controller-&gt;getPosition(&amp;distance, &amp;angle);\nranging = controller-&gt;getRanging();     </pre>
【引数】	[out]int nextScoreTable[4]
【返値】	無し

## イベントリスト

イベント名	内容
E_REACH	目標に到達した
E_CHANGE_DISTANCE	走行距離が変化した
E_CHANGE_ANGLE	旋回角度が変化した
E_CHANGE_RANING	レーザセンサの取得値が変化した
E_UP	「W」キーが押された
E_DOWN	「S」キーが押された
E_LEFT	「A」キーが押された
E_RIGHT	「D」キーが押された

※灰色のイベントは競技会では使用をしないでください