大屏交互版简单sdk 说明

###输出数据结构:

struct touch\_info {

uint64\_t stamp; /\*\*< 当前触点时间戳. \*/

uint16\_t touchid; /\*\*< 屏幕box中当前触点编号. \*/

bool isvalid; /\*\*< 当前点是否在设定的屏幕范围内. \*/

bool new\_frame; /\*\*<是否是新的一帧激光. \*/

float screen\_x; /\*\*<屏幕坐标系X(mm). \*/

float screen\_y; /\*\*<屏幕坐标系Y(mm). \*/

float laser\_x; /\*\*<雷达坐标系X(mm). \*/

float laser\_y; /\*\*<雷达坐标系Y(mm). \*/

} \_\_attribute\_\_((packed)) ;

###获取当前帧中在设定屏幕内的触点

result\_t grabScanData(touch\_info \* pointbuffer, size\_t & count, uint32\_t timeout) ;

###屏幕区域设定和雷达安装位置设定数据结构

struct LaserPose {

float x; /\*\*< 雷达在屏幕坐标系中的X位置(mm). \*/

float y; /\*\*< 雷达在屏幕坐标系中的Y位置(mm). \*/

float theta; /\*\*< 雷达在屏幕坐标系中的朝向(度). \*/

bool reversion; /\*\*< 雷达表面朝向(false: 朝外, true:朝里). \*/

};

/\*\*

\* @brief 设置雷达在屏幕坐标系中的位置 \n

\* @return 无

\* @param[in] LaserPose 雷达安装位置

\*/

void setLaserPose(const LaserPose& pose);

/\*\*

\* @brief 设置屏幕所在区域大小 \n

\* @return 无

\* @param[in] max\_x 屏幕最大X轴值

\* @param[in] max\_y 屏幕最大Y轴值

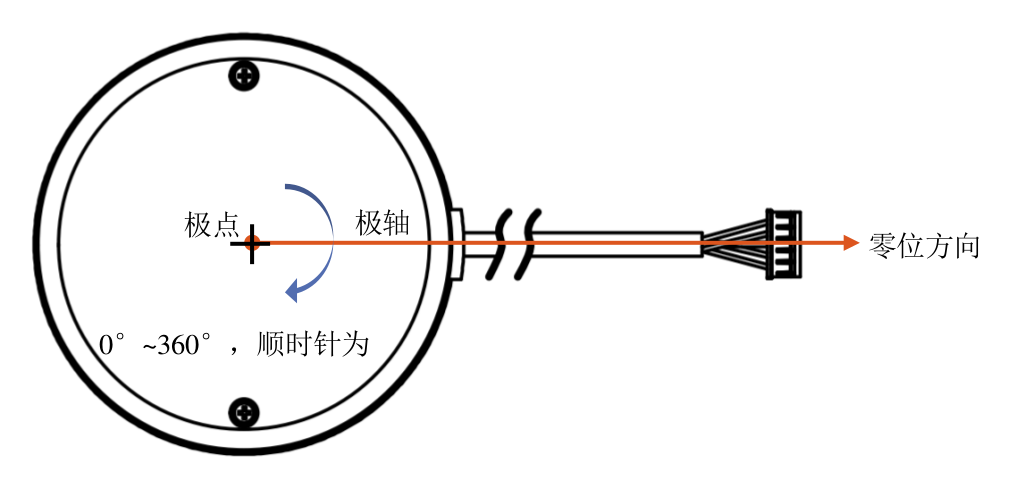
\* @param[in] min\_x 屏幕最小X轴值

\* @param[in] min\_y 屏幕最小Y轴值

\* @note左上角是屏幕坐标原点, X轴右, Y轴朝下

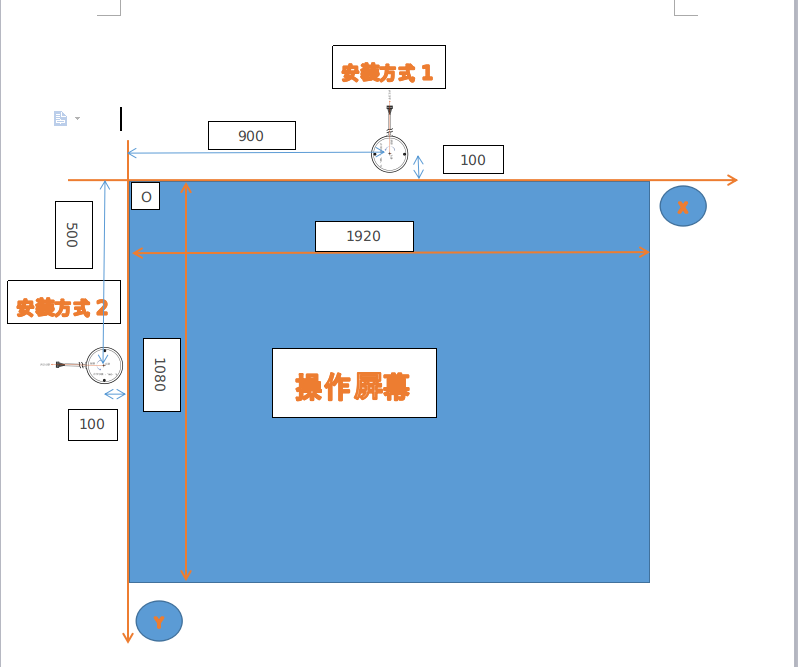
\*/

void setScreenBox(const float& max\_x, const float& max\_y, const float& min\_x, const float& min\_y);



x

Y



参数设定样例:

上图安装方式1:

屏幕X轴最小值 = 0mm;

屏幕 Y 轴最小值 = 0mm;

屏幕X轴最大值 = 1920mm;

屏幕Y轴最大值 = 1080mm;

雷达安装位置X值 = 900mm;

雷达安装位置Y值 = -100mm;

雷达安装角度Theta = -90;//雷达坐标系相对屏幕坐标系逆时针旋转了90度.

void setScreenBox(1920,1080,0, 0);

LaserPose pose;

pose.x = 900;

pose.y = -100;

pose.theta = -90;

pose.reversion = false; //雷达表面朝外

setpose(pose);

安装方式二:

屏幕X轴最小值 = 0mm;

屏幕 Y 轴最小值 = 0mm;

屏幕X轴最大值 = 1920mm;

屏幕Y轴最大值 = 1080mm;

雷达安装位置X值 = -100mm;

雷达安装位置Y值 = 500mm;

雷达安装角度Theta = -180;//雷达坐标系相对屏幕坐标系逆时针旋转了180度.

void setScreenBox(1920,1080,0, 0);

LaserPose pose;

pose.x = -100;

pose.y = 500;

pose.theta = -180;

pose.reversion = false; //雷达表面朝外

setpose(pose);

备注:雷达安装位置和角度根据实际情况设定, 注意实际屏幕和操作屏幕的换算比例. 如果安装角度是120度, 直接把雷达安装角度设置为120, 不用更改雷达零点.