

```

/**
 * 无人机事件监听器接口。
 * 提供了无人机连接状态、相机能力集、主服务可用性等一系列事件的监听方法。
 */
interface IAutelDroneListener {

    /**
     * 无人机连接状态改变监听方法。
     * @param connected 连接状态，true表示已连接，false表示已断开。
     * @param drone 无人机设备对象。
     * @deprecated 使用 [onDroneChangedListener] 替代，该方法提供更多的参数信息。
     */
    //无人机设备连接状态的监听器
    @Deprecated("see [onDroneChangedListener] more params")
    fun onDroneChangedListener(connected: Boolean, drone: IBaseDevice){}

    /**
     * 相机能力集获取状态监听方法。
     * @param fetched 获取状态，true表示获取成功，false表示获取失败。
     * @deprecated 使用 [onCameraAbilityFetchListener] 替代，该方法提供更多的参数信息。
     */
    //相机能力集变更通知
    @Deprecated("see [onCameraAbilityFetchListener] more params")
    fun onCameraAbilityFetchListener(fetched: Boolean){}

    /**
     * 主服务可用性状态监听方法。
     * @param valid 服务是否可用，true表示可用，false表示不可用。
     * @param drone 无人机设备对象。
     * @deprecated 使用 [onMainServiceValid] 替代，该方法提供了更精确的设备类型。
     */
    //主服务是否可用
    @Deprecated("onMainServiceValid param with IAutelDroneDevice")
    fun onMainServiceValid(valid: Boolean, drone: IBaseDevice){}

    /**
     * 主服务可用性状态监听方法。
     * @param valid 服务是否可用，true表示可用，false表示不可用。
     * @param drone 无人机设备对象。
     */
    /**
     * 主服务是否可用
     * @param valid true 可用，false 不可用
     * @param drone drone device
     */
    fun onMainServiceValid(valid: Boolean, drone: IAutelDroneDevice) {}

    /**
     * 相机能力集获取状态监听方法。
     * @param localFetched 本地相机能力集是否成功获取。
     * @param remoteFetched 远程相机能力集是否成功获取。
     * @param drone 无人机设备对象。
     */
}

```

```

        * 相机能力集变更通知
        * @param localFetched true 本地能力集解析成功
        * @param remoteFetched true 远程能力集解析成功
        */
    fun onCameraAbilityFetchListener(localFetched: Boolean, remoteFetched: Boolean, drone: IAutelDroneDevice){}

    /**
     * SDK业务接口调用错误监听方法。
     * @param errorInfo 错误信息对象，包含具体的错误详情。
     */
    /**
     * SDK业务接口报错回调
     * @param errorInfo
     */
    fun onSDKErrorListener(errorInfo: SDKErrorUtil) {}

    /**
     * 无人机设备创建监听方法。
     * @param drone 创建的无人机设备对象。
     */
    /**
     * 组网设备创建
     * @param drone drone device
     */
    fun onDroneCreate(drone: IAutelDroneDevice){}

    /**
     * 无人机设备销毁监听方法。
     * @param drone 销毁的无人机设备对象。
     */
    /**
     * 组网设备销毁
     * @param drone drone device
     */
    fun onDroneDestroy(drone: IAutelDroneDevice){}

    /**
     * 无人机组网/对频模式变化监听方法。
     * @param mode 变化后的模式，可以是点对点或组网模式。
     */
    /**
     * 组网/对频模式变化
     * @param mode 点对点或者组网
     */
    fun onModemModeChange(mode: ModemModeEnum){}
}

/**
 * 通信模式枚举类，定义了不同的通信模式。
 */
enum class ModemModeEnum(var value: Int) {

    /**
     * 点对点主从模式。
     * 该模式下，设备之间直接通信，分为主设备和从设备。
     */
}

```

```

P2P_MASTER_SLAVE_MODE(1),

/**
 * 组网模式。
 * 该模式下，多个设备通过自组织网络进行通信。
 */
MESH_MODE(3),

/**
 * 未知模式。
 * 用于表示无法识别的通信模式。
 */
UNKNOWN(4);

/**
 * 根据值查找对应的枚举项。
 *
 * @param value 枚举项的值。
 * @return 对应的枚举项，如果找不到则返回UNKNOWN。
 */
companion object {
    fun findEnum(value: Int): ModemModeEnum {
        val array = ModemModeEnum.values()
        for (type in array) {
            if (type.value == value) {
                return type
            }
        }
        return UNKNOWN
    }
}
}

```

```

/**
 * 飞行器设备接口，继承自IBaseDevice，提供了访问飞行器状态、管理器和其他相关信息的方法。
 */
interface IAutelDroneDevice : IBaseDevice {

    /**
     * 获取状态机信息。
     * @deprecated 已被getDeviceStateData替代。
     * @return 状态机信息。
     */
    @Deprecated("替换接口 getDeviceStateData")
    fun getStateMachine(): DroneStateMachineBean

    /**
     * 获取相册管理器。
     * @return 相册管理器实例。
     */
    fun getAlbumManager(): IAlbumManager

    /**
     * 获取航点任务管理器。

```

```

    * @return 航点任务管理器实例。
    */
fun getWayPointMissionManager(): IMissionManager

/**
 * 获取任务追踪管理器。
 * @return 任务追踪管理器实例。
 */
fun getTrackMissionManager(): ITrackMissionManager

/**
 * 获取飞机能力集管理器。
 * @return 飞机能力集管理器实例。
 */
fun getAbilitySetManager(): IAbilitySetManager

/**
 * 获取相机能力集管理器。
 * @return 相机能力集管理器实例。
 */
fun getCameraAbilitySetManger(): ICameraAbilitySetManager

/**
 * 获取精准复拍管理器。
 * @return 精准复拍管理器实例。
 */
fun getAccurateRetakeManager() : IAccurateRetakeManager

/**
 * 获取当前设备的云台类型。
 * @return 云台类型枚举。
 */
fun getGimbalDeviceType(): GimbalTypeEnum

/**
 * 获取播放器管理器。
 * @return 播放器管理器实例。
 */
fun getAutelPlayerManager(): AutelPlayerManager

/**
 * 获取文件服务管理器。
 * @return 文件服务管理器实例。
 */
fun getFileServiceManager(): FileServiceManager

/**
 * 获取设备的当前状态数据。
 * @return 设备状态数据。
 */
fun getDeviceStateData(): DroneStateData

/**
 * 获取设备的OTA管理器。
 */
fun getDeviceOTAManager()

/**

```

```

    * 获取设备能力开关集合管理器。
    */
fun getDeviceCapabilitiesManager()

/**
    * 获取RTK管理器。
    * @return RTK管理器实例。
    */
fun getRtkManager(): IRTKManager

/**
    * 获取ADS-B管理器（仅接收数据版本）。
    * @return ADS-B管理器实例。
    */
fun getAdsbManager(): IADSBManager

/**
    * 检查设备是否已连接。
    * @return 如果设备已连接返回true，否则返回false。
    */
fun isConnected(): Boolean {
    return true
}

/**
    * 检查飞机是否处于起飞准备就绪状态。
    * @return 如果飞机可以起飞返回true，否则返回false。
    */
fun isPreFlightOK(): Boolean

/**
    * 获取飞机类型。
    * @return 飞机类型的枚举值。
    */
fun getDroneType(): DroneType

/**
    * 获取飞机返回的原始飞机类型，用于升级。
    * @return 原始飞机类型的字符串表示。
    */
fun getProductDroneType(): String? {return ""}

/**
    * 检查飞机是否受控。
    * @return 如果飞机受控返回true，否则返回false。
    */
fun isControlled(): Boolean

/**
    * 获取节点ID。
    * @return 节点ID，可能为null。
    */
fun getNodeId(): Int?

/**
    * 获取组ID。
    * @return 组ID。
    */

```

```

fun getGroupId():Int

/**
 * 检查设备是否处于被监控状态。
 * @return 如果设备被监控返回true，否则返回false。
 */
fun iswatched(): Boolean

/**
 * 检查设备是否为长机。
 * @return 如果设备是长机返回true，否则返回false。
 */
fun isLeader():Boolean

/**
 * 检查设备是否为中继节点。
 * @return 如果设备是中继节点返回true，否则返回false。
 */
fun isCenter():Boolean

/**
 * 检查设备是否未加入任何群组。
 * @return 如果设备未加入任何群组返回true，否则返回false。
 */
fun issingle():Boolean

/**
 * 获取当前飞机的IP地址。
 * @return 飞机的IP地址。
 */
fun getIp(): String { return "" }

/**
 * 获取相册服务的端口号。
 * @return 相册服务的端口号。
 */
fun getAlbumPort(): Int { return 0}

/**
 * 获取设备名称，用于组网。
 * @return 设备名称。
 */
fun getName(): String { return "" }

/**
 * 获取文件服务的端口号。
 * @return 文件服务的端口号。
 */
fun getFileServicePort(): Int {return 0}

/**
 * 获取相册基础URL。
 * @return 相册基础URL。
 */
fun getAlbumBaseUrl():String{ return "" }

/**
 * 获取进度查询URL。

```

```

        * @return 进度查询URL。
    */
    fun getProgressQueryUrl():String{ return "" }

    /**
     * 获取OTA上传URL。
     * @return OTA上传URL。
     */
    fun getOtaUploadUrl():String{ return "" }

    /**
     * 获取任务上传URL。
     * @return 任务上传URL。
     */
    fun getMissionUploadUrl():String{ return ""}

    /**
     * 获取文件基础URL。
     * @return 文件基础URL。
     */
    fun getFileBaseUrl(): String {return ""}

    /**
     * 生成示例字符串。
     * @return 生成的示例字符串。
     */
    fun toSampleString():String

    /**
     * 获取飞机的SN号。
     * @return 飞机的SN号，可能为null。
     */
    fun getDroneSn(): String? {
        return null
    }

    /**
     * 获取飞机的版本号。
     * @return 飞机的版本号，可能为null。
     */
    fun getDroneVersion(): String? {
        return null
    }
}

```