**AIV Program API文档说明**

**一、功能API**

1.1、获取操作的结果, 返回Int整形结果

/// <summary>

/// 获取signal对应操作的结果。在setSignal返回后，调用此API

/// </summary>

/// <returns></returns>

int signal.GetSignalOperationIntResult();

string signal.GetSignalOperationStringResult();

1.2、Signal 操作

设置信号SetSignal提供了两种形式的API,传一个参数或多个参数。其中signalName是信号标识，匹配ini配置文件。传递多个参数API除signalName外，其余参数可根据需求配置使用。WaitSignal为等待信号。

signal.SetSignal(signalName);

例：

第三方ini文件配置：signal\_261=1:626:1 #QDSC AGV1 顶升到位通知；

Server方ini文件配置：Signal\_AGV1\_OpConfirm1=2:261:1:1 #顶升到位通知

AIV Program： Signal.SetSignal(Signal\_AGV1\_OpConfirm1)

signal.SetSignal(signalName，2,3,4);

例：将第二个参数用为AGV所处当前位置，第三和第四个参数暂时没使用缺省为1。

第三方ini文件配置：signal\_71=1:622:1:2 #agv1 Signal\_Spec31

Server方ini文件配置：signalArrive1=2:71:1:1 ## arrive

AIV Program： Signal.SetSignal(signalArrive1,2,1,1)

signal.WaitSignal(signalName);

1. AIV 小车充电控制

2.1 /// <summary>

/// 请求充电 -- 设置aiv的继电器连接上

/// </summary>

/// <returns></returns>

bool aiv\_charge.ChargeReady();

2.2 /// <summary>

/// 充电完成--设置aiv断开内部继电器。

/// </summary>

/// <returns></returns>

bool aiv\_charge.ChargeFinished();

2.3 /// <summary>

///

/// </summary>

/// <returns></returns>

bool aiv\_battery.IsLowPower()

2.4 /// <summary>

/// 获取当前点的充电桩序号 -- 根据当前的workpoint id。

/// </summary>

/// <param name="workPointId"></param>

/// <returns></returns>

int aiv\_battery.GetChargingDeviceID(int workPointId);

2.5//充电桩

//chargingMsgType = 1; // 充电命令类型值

//chargeReqMsgId = 2; // 请求充电桩伸出充电头的消息ID

//chargeFinishMsgId = 3; // 请求充电桩收回充电头的消息ID

//string signal.GetSignalName(int signalType, int nSignalId, int nSN);

2.6请求充电API

/// <summary>

/// 请求充电

/// </summary>

/// <param name="powerCap">电池电量</param>

/// <returns></returns>

task.RequestCharging(double powerCap);

TASK

3.1. 关于aiv参与工作的操作通知API：

/// <summary>

/// 2017-06-14 add an API

/// task service need implement this API.

/// Online Service Component doesn't need implement this API.

/// </summary>

/// <param name="operationSignal"></param>

/// <returns></returns>

bool StartOperation(string operationSignal);

bool FinishOperation(string operationSignal);

3.2获取下一工作点许可

PointGrantStatus ITaskDispatchService.GetNextTaskPointPermission(int taskPointId)

void GetNextTaskPointPermissionUntillGrant(int taskPoint);

PointGrantStatus: {

Refused = 0,

Granted = 1,

Pending = 2,

WAITING ,

}

sample:

// 直到granted 才返回。

GetNextTaskPointPermissionUntillGrant(nextPointId);

PointGrantStatus = taskresult.ResultOnPointPermission()

or:

nextPermission = task.GetNextTaskPointPermission(nextPointId);

// 直到granted（1）

while (nextPermission ~= 1) do

aiv.WaitTimeout(1000);

nextPermission = taskresult.ResultOnPointPermission()

end

3.3.aiv.EnableOriginPoint(bool)

3.4 .laser设定安全模式的api：

laser.SetSecurityLaserMode(int nMode)

RFID

4.1.获取当前RFID

string RFID.GetCurrentRFID()

4.2. 更新设置直行RFID API:

RFID.SetStraightControlRFID("rfid")

4.3. 添加设置通过下个rfid后的速度

RFID.SetNextRFIDSpeed("rfid",speed);

\*\* 使用时，需要PLC 程序更新匹配。

4.4等待读取RFID id为等待读取编号

id1 = "0020"

bResult=RFID.WaitRFID(id1, fal se)

4.5设置RFID控制API

示例：

RFID.SetStopControlRFID("0308");

RFID.SetLeftControlRFID("0402");

RFID.SetRightControlRFID("2944");

动作API

5.1PLC控制转角的API

aiv.SetTurnThetaCommand(theta);

更新了API： aiv.SetTurnThetaCommand ：

添加了maxW -- 设置转角的角速度。

aiv.SetTurnThetaCommand(double theta, double maxW=800.0)

5.2完成磁导的divergence路径选择api：

/// <summary>

/// 磁条导航中转左路径

/// </summary>

void TurnToLeftPathByPermeameter();

5.3

/// <summary>

/// 磁条导航中转右路径

/// </summary>

void TurnToRightPathByPermeameter();

5.4

/// <summary>

/// 磁条分叉路口中直行路径

/// </summary>

void TurnStraightPathByParmeameter();

5.5 停止API：void Stop();

5.6. aiv.MoveWithSpeedByPermeameter(800, true)

5.7 aiv.TurnThetaWithWheel(double theta, bool turnLeft);

5.8. void TurnTheta(double theta,double w,bool leftTurn); // 使用时，请先关闭磁导模式。

5.9. void MoveToPointAbsolute(double s, double x, double y, double theta, double maxVelocity);

5.10为aiv添加顶升/滚筒操作API：

/// <summary>

/// 2018-01-03 scroll API

/// </summary>

/// <param name="scrollType">1: scroll forward, 2-- scroll backward.</param>

/// <returns></returns>

int Scroll(int scrollType)

AIV\_server\_2018-02-12：

更新scroll api，添加超时处理：如果超时30s还没有到位，则返回-1；到位则返回1；

1. 添加broadcast aiv信息处理，在配置文件中添加了配置项：

-------------------

; 广播数据间隔

[broadcast]

interval = 20

-------------------

配合Online调度系统，如果Online需要device-info信息频次快慢，调整设置interval值。

**六、日志**

1.为scirpt添加logger对象，可以记录日志了。

/// <summary>

///

/// </summary>

/// <param name="strInfo"></param>

void Info(string strInfo);

/// <summary>

///

/// </summary>

/// <param name="strWarn"></param>

void Warn(string strWarn);

/// <summary>

///

/// </summary>

/// <param name="strError"></param>

void Error(string strError);

aiv-prgram的脚本日志命名script.log