

ICS 35.100.01

CCS M19

T/CTS

中国道路交通安全协会团体标准

T/CTS XXXX—2025

基于物联网的道路交通安全管控设备通信 协议第2部分：交通信号控制机

Road traffic safety equipment communication protocol based on Internet of Things

Part 2: Traffic signal controller

（征求意见稿）

2025 - XX - XX发布

2025- XX - XX实施

中国道路交通安全协会 发布

目次

前 言..... III

1 范围..... 1

2 规范性引用文件..... 1

3 术语和定义..... 1

4 技术要求..... 2

 4.1 一般规定..... 2

 4.2 数据帧内容..... 2

 4.3 数据值MIB..... 20

附 录 A..... 116

附 录 B..... 120

 B.1 通信规程..... 120

 B.2 用户访问授权..... 120

 B.3 单台信号机控制多路口流程..... 120

 B.4 总则定义数据帧报文示例..... 120

 B.5 JSON 报文示例..... 122

参 考 文 献..... 126

前 言

本文件按照GB/T1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

T/CTS XXXX《基于物联网的道路交通安全设备通信协议》分为以下部分：

- 第1 部分：总则
- 第2 部分：交通信号控制机
- 第3 部分：LED道路交通诱导可变信息标志
- 第4 部分：可变交通标志
- 第5 部分：交通检测器
- 第6 部分：道路环境检测器
- 第7 部分：智能运维机柜
- 第8 部分：交通安全警示设施
- 第9 部分：车路协同路侧单元
- 第10 部分：车路协同车载单元
- 第11 部分：平台间控制互联与数据交换

本文件为T/CTS XXXX的第2部分：交通信号控制机。

本文件可能涉及相关专利，鼓励组织和个人披露所拥有和知晓的必要专利。本文件的发布机构不承担识别这些专利的责任。

本文件由中国道路交通安全协会提出并归口。

本文件起草单位：

本文件主要起草人：

基于物联网的道路交通安全设备通信协议

第2部分：交通信号控制机

1 范围

本文件规定了交通信号控制机与集中控制器或上位机进行数据通信的数据帧格式内容。
本文件适用于物联网条件下道路交通安全设备中交通信号控制机与集中控制器或上位机间的数据通信。

2 规范性引用文件

数据帧结构见 T/CTS XXXX.1—2024《基于物联网的道路交通安全管控设备通信协议第1部分 总则》。

3 术语和定义

3.1

交通信号控制机 road traffic signal controller

能够改变道路交通信号顺序、调节配时并能控制道路交通信号灯运行的装置。

3.2

信号灯组 signal light group

一个完整的车辆红、黄、绿三单元灯或行人红、绿二单元灯的组合。

3.3

信号通道 signal channel

信号机中，用于控制信号灯组输出的单元或装置。通常一个信号通道提供车辆红黄绿三个灯色驱动或者行人红绿两个灯色驱动。

注：通常一个信号通道可以驱动一个或者多个信号灯组，但是目前也存在一些多功能信号灯组，需要多个信号通道驱动一个多功能信号灯组，因此信号通道和信号灯组不是一一对应关系。

3.4

相位 phase

同时获得且同时失去通行权的一股或多股交通流所对应的信号序列。

3.5

相位差 offset

协调控制中，交叉口相位协调点与指定的基准时间或指定交叉口的相位协调点之间的时间差。

注1：相位协调点可以为协调相位的绿灯开始、绿灯结束、红灯开始的时刻。

注2：相位差分为绝对相位差和相对相位差两类。绝对相位差是指交叉口相位协调点与指定的基准时间之间的时间

差；相对相位差是指交叉口相位协调点与指定交叉口相位协调点之间的时间差。

注3：通常将除以交叉口周期所得到的余数作为相位差的规范化取值。

3.6

冲突 traffic conflict

交通流之间其通行轨迹相互交叉、合流或分流的交通现象。

3.7

绿灯间隔时间 inter-green time

一个相位绿灯时间结束至下一个相位绿灯时间开始之间的间隔时间。

3.8

阶段 stage; signal stage

信号周期内，一个或多个相位同时获得路权的状态。

3.9

阶段过渡约束 phase transition constraints
阶段进行转换的约束规则。

3.10

阶段链 phase stage chain
信号周期内，一组阶段的出现序列。

3.11

配时方案 intersection signal timing plan; intersection signal timing pattern
交叉口关于相位设置、相位序列设置、相位时长等参数的有序集合。

3.12

日计划 day plan; time-of-day plan
为适应不同交通需求而设置的一天中信号控制时间段、控制方式及方案的集合。

3.13

调度表 date schedule
信号机内表示日期与所用日计划的对应关系的表格。

3.14

本地控制 local control
信号机本地路口的自主控制方式。

3.15

中心控制 center control
上位机向信号机发送命令控制信号机运行的方式。

4 技术要求

4.1 一般规定

数据帧结构见 T/CTS XXXX.1—2024《基于物联网的道路交通安全设备通信协议 第1部分 总则》。

4.2 数据帧内容

4.2.1 数据帧标识号

- a) 状态标记：M(n)，M为应支持项，(n)表示有多个元素时，表示不同的元素编号；O为可选项；
- b) 类型标记：Q 为支持查询协议；S 为支持设置协议；T 为支持主动上报协议；
- c) 标识号等级数量根据控制对象需求存在多种等级；
- d) 数据类详细数据属性定义见相关章节的标识号MIB定义；
- e) 在JSON和XML数据格式中对象的键与MIB定义的对象名称一致；
- f) 当使用SNMP协议访问本协议定义的交通信号控制对象时，设备标识号完整路径为：iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.设备.交通信号控制机.A.B.C... (1.3.6.1.4.1.61332.3.2.2.A.B.C...)，A、B、C、...对应具体的设备标识号，详见本系列协议中相应数据类部分描述。

4.2.2 数据值

4.2.2.1 数据类1 通用数据

数据类1定义为所有类型设备通用的数据，标识号为1.x，详细定义见总则。

4.2.2.2 数据类2 基础信息

数据类2定义为基本信息，详见表1。

表1 基本信息

标识号				状态标记	类型标记
2.基础信息 (basic)	1.信号机编号(controllerID)			M	Q、S
	2.机柜编码(cabinetID)			O	Q、S
	3.信号机控制路口数量(intersectionNum)			M	Q、S
	3.信号机控制路口属性表(intersectionInfoTable)	1.控制路口编号(index)	n.对象序号	M(n)	Q
		2.控制路口全局编号(globalID)	n.对象序号	M(n)	Q、S
		3.控制路口访问端口(comPort)	n.对象序号	M(n)	Q、S
	4.信号语义(semanticCfg)	1.支持语义类型(supportSemantic)		M	Q
		2.当前应用语义类型(useSemantic)		M	Q、S
	5.时间操作配置(timeCfg)	1.时间计算基准(timeDatum)		M	Q、S
		2.时间获取方式(timeGetMethod)		M	Q、S
		3.卫星时钟标志(GNSSClock)		O	Q
	6.上位机IPV4网络配置(managerIPV4Cfg)	1.IP地址(IP)		O	Q、S
		2.通信端口(comPort)		O	Q、S
		3.通信类型(comType)		O	Q、S
	7.上位机IPV6网络配置(managerIPV6Cfg)	1.IP地址(IP)		O	Q、S
		2.通信端口(comPort)		O	Q、S
		3.通信类型(comType)		O	Q、S

标识号示例

- a) 信号机编号的标识号：2.1，其中第一字段2表示一级标识号，为基本信息；第二字段1表示二级标识号，为信号机编号。
- b) 支持语义类型的标识号：2.4.1，其中第一字段2表示一级标识号，为基本信息；第二字段4表示二级标识号，为信号语义；第三字段1表示三级标识号，为支持语义类型。
- c) 信号机控制路口4的全局编号的标识号：2.3.2.4，其中第一字段2表示一级标识号，为基本信息；第二字段3表示二级标识号，为信号机控制路口属性；第三字段2表示三级标识号，为控制路口全局编号；第四字段4表示四级标识号，为第4个控制路口。

4.2.2.3 数据类3 通道灯组信息

数据类3定义为通道信息，详见表2。

- a) 通信协议约定：信号机的最大通道数为64；
- b) 通道配置表、通道状态表、通道控制表中的数据通过统一编码的通道编号相关联。

表2 通道信息

标识号				状态标记	类型标记
3.通道信息 (channel)	1.实际通道数(maxChannel)			M	Q
	2.通道配置表	1.通道编号(index)	n.对象序号	M(n)	Q

标识号				状态标记	类型标记
	(channelCfgTable)	2.通道灯组类型(type)	n.对象序号	M(n)	Q、S
		3.通道归属路口(intersectionIndex)	n.对象序号	M(n)	Q、S
		4.通道对应交通流的方向(direction)	n.对象序号	O(n)	Q、S
		5.通道对应交通流的流向(movement)	n.对象序号	O(n)	Q、S
		6.通道对应的车道(laneID)	n.对象序号	O(n)	Q、S
		7.交通流放行规则(greenType)	n.对象序号	O(n)	Q、S
		8.许可放行关联通道(greenIncluded)	n.对象序号	O(n)	Q、S
		9.保护放行切换许可放行灯组动作(greenAction)	n.对象序号	O(n)	Q、S
		10.通道灯组ID(V2I_ID)	n.对象序号	O(n)	Q、S
		3.通道状态表(channelStateTable)	1.通道编号(index)	n.对象序号	M(n)
	2.通道灯组状态(SignalState)		n.对象序号	M(n)	Q、T
	3.交通流放行状态(movementState)		n.对象序号	O(n)	Q、T
	4.通道控制表(channelCtlTable)	1.通道编号(index)	n.对象序号	M(n)	Q
		2.通道屏蔽(mask)	n.对象序号	M(n)	Q、S
		3.通道禁止(disable)	n.对象序号	M(n)	Q、S

标识号示例：

- a) 实际通道数的标识号：3.1，其中第一字段3表示一级标识号，为通道信息；第二字段1表示二级标识号，为实际通道数。
- b) 通道4灯组状态的标识号：3.3.2.4，其中第一字段3表示一级标识号，为通道信息；第二字段3表示二级标识号，为通道状态表；第三字段2表示三级标识号，为通道灯组状态；第四字段4表示四级标识号，为第4通道。

4.2.2.4 数据类4 相位信息

数据类4定义为相位信息，详见表3。

- a) 通信协议规定：单个路口最多包含64个相位；
- b) 通信协议规定：单个相位最多包含64个通道；
- c) 通信协议规定：单个相位最多包含128个需求。

表3 相位信息

标识号				状态标记	类型标记
4.相位信息 (phase)	1.实际相位数 (maxPhases)			M	Q
	2.相位配置表 (phaseCfgTable)	1.相位编号(index)	n.对象序号	M(n)	Q
		2.相位的通道(channelIncluded)	n.对象序号	M(n)	Q、S
		3.取消路权过渡灯色1类型 (onLoseStep1LightType)	n.对象序号	M(n)	Q、S
		4. 取消路权过渡灯色1时间 (onLoseStep1Time)	n.对象序号	M(n)	Q、S
		5. 取消路权过渡灯色2类型 (onLoseStep2LightType)	n.对象序号	M(n)	Q、S

标识号			状态标记	类型标记	
		6. 取消路权过渡灯色2时间 (onLoseStep2Time)	n.对象序号	M(n)	Q、S
		7. 取消路权过渡灯色3类型 (onLoseStep3LightType)	n.对象序号	M(n)	Q、S
		8. 取消路权过渡灯色3时间 (onLoseStep3Time)	n.对象序号	M(n)	Q、S
		9. 获得路权过渡灯色1类型 (onGetStep1LightType)	n.对象序号	M(n)	Q、S
		10. 获得路权过渡灯色1时间 (onGetStep1Time)	n.对象序号	M(n)	Q、S
		11. 获得路权过渡灯色2类型 (onGetStep2LightType)	n.对象序号	M(n)	Q、S
		12. 获得路权过渡灯色2时间 (onGetStep2Time)	n.对象序号	M(n)	Q、S
		13. 获得路权过渡灯色3类型 (onGetStep3LightType)	n.对象序号	M(n)	Q、S
		14. 获得路权过渡灯色3时间 (onGetStep3Time)	n.对象序号	M(n)	Q、S
		15. 开机获得路权灯色1类型 (onStartupGetStep1LightType)	n.对象序号	M(n)	Q、S
		16. 开机获得路权灯色1时间 (onStartupGetStep1Time)	n.对象序号	M(n)	Q、S
		17. 开机获得路权灯色2类型 (onStartupGetStep2LightType)	n.对象序号	M(n)	Q、S
		18. 开机获得路权灯色2时间 (onStartupGetStep2Time)	n.对象序号	M(n)	Q、S
		19. 开机获得路权灯色3类型 (onStartupGetStep3LightType)	n.对象序号	M(n)	Q、S
		20. 开机获得路权灯色3时间 (onStartupGetStep3Time)	n.对象序号	M(n)	Q、S
		21. 开机取消路权灯色1类型 (onStartupLoseStep1LightType)	n.对象序号	M(n)	Q、S
		22. 开机取消路权灯色1时间 (onStartupLoseStep1Time)	n.对象序号	M(n)	Q、S
		23. 开机取消路权灯色2类型 (onStartupLoseStep2LightType)	n.对象序号	M(n)	Q、S
		24. 开机取消路权灯色2时间 (onStartupLoseStep2Time)	n.对象序号	M(n)	Q、S
		25. 开机取消路权灯色3类型 (onStartupLoseStep3LightType)	n.对象序号	M(n)	Q、S
		26. 开机取消路权灯色3时间 (onStartupLoseStep3Time)	n.对象序号	M(n)	Q、S
		27. 相位最小绿时间(minGreen)	n.对象序号	M(n)	Q、S
		28. 相位最大绿时间1(maxGreen1)	n.对象序号	M(n)	Q、S
		29. 相位最大绿时间2(maxGreen2)	n.对象序号	O(n)	Q、S
		30. 相位延长绿时间 (extendedGreen)	n.对象序号	M(n)	Q、S
		31. 相位的需求(phaseRequest)	n.对象序号	M(n)	Q、S

标识号				状态标记	类型标记
		32.相位最大绿时间3(maxGreen3)	n.对象序号	M(n)	Q、S
	3.相位控制表 (phaseCtlTable)	1.相位编号(index)	n.对象序号	O(n)	Q
		2.相位屏蔽(mask)	n.对象序号	O(n)	Q、S
		3.相位禁止(disable)	n.对象序号	O(n)	Q、S
	4.相位状态表 (phaseStateTable)	1.相位编号(index)	n.对象序号	M(n)	Q
		2.相位状态(state)	n.对象序号	M(n)	Q
		3.相位已经运行时间 (elapsedTime)	n.对象序号	M(n)	Q
		4.相位剩余时间(remainderTime)	n.对象序号	O(n)	Q

标识号示例：

- a) 实际相位数的标识号：4.1，其中第一字段4表示一级标识号，为相位信息；第二字段1表示二级标识号，为实际相位数。
- b) 相位8的通道的标识号：4.2.2.8，其中第一字段4表示一级标识号，为相位信息；第二字段2表示二级标识号，为相位配置表；第三字段2表示三级标识号，为相位的通道；第四字段8表示四级标识号，为第8相位。

4.2.2.5 数据类5 检测器信息

数据类5定义为检测器信息，详见表4。

- a) 通信协议规定信号机最大检测器数为128；
- b) 检测器配置表和检测器数据表通过检测器编号相关联；
- c) 对于能够用简单0、1描述的检测数据，检测器与相位的关联关系通过相位的需求实现关联；对于无法用0、1数据描述的检测数据，检测器与相位的关联关系通过检测器归属的车道号以及相位的通道一通道的车道实现相互关联。

表4 检测器信息

标识号				状态标记	类型标记
5.检测器信息 (detector)	1.实际检测器数 (maxDetector)			M	Q
	2.检测器配置表 (detectorCfgTable)	1.检测器编号(index)	n.对象序号	M(n)	Q
		2.检测器类型(type)	n.对象序号	M(n)	Q、S
		3.检测器采集周期1/流量采集周期(samplePeriod1)	n.对象序号	M(n)	Q、S
		4.检测器采集周期2/占有率采集周期(samplePeriod2)	n.对象序号	M(n)	Q、S
		5.检测器安装位置描述(info)	n.对象序号	O(n)	Q、S
		6.检测器归属路口序号(intersectionIndex)	n.对象序号	O(n)	Q、S
		7.检测器归属车道号(laneID)	n.对象序号	O(n)	Q、S
		8.检测器检测位置(position)	n.对象序号	O(n)	Q、S
		9.检测器检测对象(targetType)	n.对象序号	O(n)	Q、S
		10.检测器检测数据类型(dataType)	n.对象序号	O(n)	Q、S
		11.检测器故障判断有效(DFMEnable)	n.对象序号	O(n)	Q、S
		12.检测器常闭故障时间	n.对象序号	O(n)	Q、S

标识号				状态标记	类型标记
		(DFMonTime)			
		12.检测器常开故障时间 (DFMoffTime)	n.对象序号	O(n)	Q、S
	3.检测器数据状态表 (detectorDataTable)	1.检测器编号(index)	n.对象序号	M(n)	Q
		2.存在状态/行人请求状态(state)	n.对象序号	M(n)	Q、T
		3.检测状态数据生成时间 (sampleTime)	n.对象序号	M(n)	Q、T
		4.检测对象的速度(targetSpeed)	n.对象序号	O(n)	Q、T
		5.检测对象的类型(targetType)	n.对象序号	O(n)	Q、T
		6.检测对象的身份(targetID)	n.对象序号	O(n)	Q、T
		7.排队长度(queeneLength)	n.对象序号	O(n)	Q、T
		8.排队数量/等候数量/行人数量 (targetNum)	n.对象序号	O(n)	Q、T
		9.检测器区域目标总数 (targetTotalNum)	n.对象序号	O(n)	Q、T
		10.检测器区域 目标列表(targetDataTable)	1.目标编号 (index)	m.n.对象 序号	O(n)
		10.检测器区域 目标列表 (targetDataTable)	2.目标类型 (type)	m.n.对象序 号	O(n)
			3.目标速度 (speed)	m.n.对象序 号	O(n)
			4.目标位置 (position)	m.n.对象序 号	O(n)
		1.检测器编号 (index)	n.对象序号	O(n)	Q、T
	4.检测器控制表 (detectorCtlTable)	2.检测器开关(enable)	n.对象序号	O(n)	Q、S
		3.检测器重启(restart)	n.对象序号	O(n)	Q、S
		4.检测器数据主动上传使能 (trapEnable)	n.对象序号	O(n)	Q、S
		5.检测器数据主动上传周期 (trapPeriod)	n.对象序号	O(n)	Q、S
		6.检测器状态的强制置位			

标识号示例：

- a) 实际检测器数的标识号： 5.1，其中第一字段5表示一级标识号，为检测器信息；第二字段1表示二级标识号，具体为实际检测器数。
- b) 第7检测器类型的标识号： 5.2.2.7，其中第一字段5表示一级标识号，为检测器信息；第二字段2表示二级标识号，具体为检测器配置表；第三字段2表示三级标识号，具体为检测器类型；第四字段7表示四级标识号，具体为第7检测器。
- c) 第10检测器第100个检测目标的位置的标识号： 5.3.10.4.100.10，其中第一字段5表示一级标识号，为检测器信息；第二字段3表示二级标识号，具体为检测器数据状态表；第三字段10表示三级标识号，具体为检测器区域目标列表；第四字段4表示四级标识号，具体为目标位置；第五字节100表示五级标识号，具体为目标位置的100个元素，即第100个检测目标的目标位置；第六字节10表示六级标识号，具体为检测器数据状态表的第10个元素，即第10检测器的状态。

4.2.2.6 数据类6 阶段信息

数据类6定义为阶段信息，详见表5。

当交通信号机采用阶段语义运行时，本数据类实际生效。当采用环语义时，本数据类不生效，任何对本数据类的访问都将返回无访问权限错误。

- a) 通信协议规定：信号机最大阶段数为64，编号为：1~64；
b) 通信协议约定：单个阶段最多包含64个相位。

表5 阶段信息

标识号				状态标记	类型标记
6.阶段信息 (stage)	1.实际配置阶段数 (maxStage)			M	Q
	2.阶段配置表 (stageCfgTable)	1.阶段编号(index)	n.对象序号	M(n)	Q
		2.阶段的相位(includedPhase)	n.对象序号	M(n)	Q、S
		3.阶段所属路口(intersectionNo)	n.对象序号	M(n)	Q、S
	3.阶段状态表 (stageStateTable)	1.阶段编号(index)	n.对象序号	M(n)	Q
		2.阶段状态(state)	n.对象序号	M(n)	Q
		3.阶段已经运行时间 (elapsedTime)	n.对象序号	M(n)	Q
		4.阶段剩余时间(remainderTime)	n.对象序号	O(n)	Q
	4.阶段控制表 (stageCtlTable)	1.阶段编号(index)	n.对象序号	O(n)	Q
		2.阶段软件需求 (softwareRequest)	n.对象序号	O(n)	Q、S
		3.阶段屏蔽标志(mask)	n.对象序号	O(n)	Q、S
		4.阶段禁止标志(disable)	n.对象序号	O(n)	Q、S

标识号示例：

- a) 实际配置的阶段数的标识号： 6.1，其中第一字段6表示一级标识号，具体为阶段信息；第二字段1表示二级标识号，具体为实际配置阶段数。
b) 阶段9的相位的标识号： 6.2.2.9，其中第一字段6表示一级标识号，具体为阶段信息；第二字段2表示二级标识号，具体为阶段配置表；第三字段2表示三级标识号，具体为阶段的相位；第四字段9表示四级标识号，具体为第9阶段。

4.2.2.7 数据类7 相位安全信息

数据类7定义为相位安全信息，详见表6。

注意本数据类中的相位冲突和相位的并发，逻辑上是同一事项的不同角度表述，理论上可以合并，但是考虑到使用习惯，还是将两者同时定义。

表6 安全信息

标识号				状态标记	类型标记
7.相位安全信息 (safe)	1.相位冲突配置表 (conflictTable)	1.序列编号(index)	n.对象序号	M(n)	Q
		2.冲突相位序列 (conflictList)	n.对象序号	M(n)	Q、S
	2.相位绿间隔配置表 (greenIntervalTable)	1.序列编号(index)	n.对象序号	M(n)	Q
		2.绿间隔时间序列 (intervalList)	n.对象序号	M(n)	Q、S
	3.相位并发配置表 (concurrentTable)	1.序列编号(index)	n.对象序号	O(n)	Q
		2.并发相位序列 (concurrentList)	n.对象序号	O(n)	Q、S
	4.相位关联配置表 (conjunctionTable)	1.序列编号(index)	n.对象序号	O(n)	Q
		2.关联相位序列	n.对象序号	O(n)	Q、S

标识号				状态标记	类型标记
		(conjunctionList)			

标识号示例：

举例说明相位1（相位冲突配置表的元素1）的所有数据标识号：

- a) 相位1的冲突相位编号标识号：7.1.2.1，其中第一字段7表示一级标识号，具体为安全信息；第二字段1表示二级标识号，具体为相位冲突配置表；第三字段2表示三级标识号，具体为冲突相位序列；第四字段1表示四级标识号，具体为相位1的序号。

4.2.2.8 数据类8 紧急优先信息

数据类8定义为紧急优先信息，详见表7。

- a) 优先配置表和优先状态表通过优先编号关联；

b) 优先响应间隔用来控制连续两次优先请求的响应时间间隔，当前一次优先被响应之后，启动内部间隔计时器，该计时器为从响应时间间隔设置开始的倒计时时钟，当倒计时为0以后可以响应下一次的优先请求；

- c) 紧急配置表和紧急状态表通过紧急编号关联；

d) 紧急响应间隔用来控制连续两次紧急请求的响应时间间隔，当前一次紧急被响应之后，启动内部间隔计时器，该计时器为从响应时间间隔设置开始的倒计时时钟，当倒计时为0以后可以响应下一次的紧急请求。

表7 紧急优先

标识号				状态标记	类型标记
8.紧急优先信息 (preempt)	1.实际优先数量 (maxPreempt)			O	Q
	2.优先配置表 (preemptCfgTable)	1.优先编号(index)	n.对象序号	O(n)	Q
		2.优先申请阶段(callStage)	n.对象序号	O(n)	Q、S
		3.优先申请相位(callPhase)	n.对象序号	O(n)	Q、S
		4.优先申请优先级(priority)	n.对象序号	O(n)	Q、S
		5.优先屏蔽标志(mask)	n.对象序号	O(n)	Q、S
		6.优先来源(source)	n.对象序号	O(n)	Q、S
		7.优先响应间隔(rspInterval)	n.对象序号	O(n)	Q、S
	3.优先状态表 (preemptStateTable)	1.优先编号(index)	n.对象序号	O(n)	Q
		2.优先申请状态(requestState)	n.对象序号	O(n)	Q
		3.优先执行状态(executeState)	n.对象序号	O(n)	Q
		4.优先间隔计时(intervalCnt)	n.对象序号	O(n)	Q
	4.实际紧急数量 (maxUrgent)			O(n)	Q
	5.紧急配置表 (urgentCfgTable)	1.紧急编号(index)	n.对象序号	O(n)	Q
		2.紧急申请阶段(callStage)	n.对象序号	O(n)	Q、S
		3.紧急申请相位(callPhase)	n.对象序号	O(n)	Q、S
		4.紧急申请优先级(priority)	n.对象序号	O(n)	Q、S
		5.紧急屏蔽标志(mask)	n.对象序号	O(n)	Q、S
		6.紧急来源(source)	n.对象序号	O(n)	Q、S
		7.紧急响应间隔(rspInterval)	n.对象序号	O(n)	Q、S
	6.紧急状态表	1.紧急编号(index)	n.对象序号	O(n)	Q

标识号			状态标记	类型标记
	(urgentStateTable)	2.紧急申请状态(requestState)	n.对象序号	O(n)
		3.紧急执行状态(executeState)	n.对象序号	O(n)
		4.紧急间隔计时(intervalCnt)	n.对象序号	O(n)
				Q

标识号示例：

- a) 实际优先数量的标识号：8.1，其中第一字段8表示一级标识号，具体为紧急优先；第二字段1表示二级标识号，具体为实际优先数量。
- b) 第6号优先申请状态的标识号：8.3.2.6，其中第一字段8表示一级标识号，具体为紧急优先；第二字段3表示二级标识号，具体为优先状态表；第三字段2表示三级标识号，具体为优先申请状态；第四字段6表示四级标识号，具体为第6号优先。

4.2.2.9 数据类9 阶段方案信息

数据类9定义为阶段方案信息，详见表8。

当交通信号机采用阶段语义运行时，本数据类实际生效。当采用环语义时，本数据类不生效，任何对本数据类的访问都将返回无访问权限错误。

- a) 通信协议规定：阶段方案数量最大为128。
- b) 通信协议规定：单个阶段方案最多包含16个阶段。

表8 阶段方案信息

标识号			状态标记	类型标记
9.阶段方案信息 (stagePattern)	1.实际方案数 (maxPattern)		M	Q
	2.方案配置表 (patternCfgTable)	1.方案编号(index)	n.对象序号	M(n)
		2.阶段所属路口(intersectionNo)	n.对象序号	M(n)
		3.方案周期(cycle)	n.对象序号	M(n)
		4.方案协调序号(coordinationIndex)	n.对象序号	O(n)
		5.方案相位差时间(offset)	n.对象序号	O(n)
		6.方案阶段链(stageChain)	n.对象序号	M(n)
		7.方案阶段时间链(stageTimeChain)	n.对象序号	M(n)
		8.方案阶段出现类型链 (stageTypeChain)	n.对象序号	M(n)
		9.方案相位差计算参考点 (offsetSyncPoint)	n.对象序号	O(n)
		10.方案阶段相位晚启动时间链 (lateBeginTimeChain)	n.对象序号	O(n)
		11.方案阶段相位早结束时间链 (earlyEndTimeChain)	n.对象序号	O(n)
		12.方案阶段最小绿灯时间链 (minGreenChain)	n.对象序号	M(n)
		13.方案阶段最大绿灯时间1链 (maxGreen1Chain)	n.对象序号	M(n)
		14.方案阶段最大绿灯时间2链 (maxGreen2Chain)	n.对象序号	M(n)
		15.方案阶段单位延长绿灯时间链 (extendedGreenChain)	n.对象序号	M(n)
		16.方案阶段相位感应链 (detectorChain)	n.对象序号	O(n)

标识号				状态标记	类型标记
		17.方案阶段最大浪费时间链(maxWasteChain)	n.对象序号	O(n)	Q、S
		18.方案阶段切换损失时间链(lossOnSwitchChain)	n.对象序号	O(n)	Q、S

标识号示例：

- a) 实际阶段方案数的标识号：9.1，其中第一字段9表示一级标识号，具体为阶段方案信息；第二字段1表示二级标识号，具体为实际阶段方案数。
- b) 阶段方案11的周期的标识号：9.2.3.11，其中第一字段9表示一级标识号，具体为阶段方案信息；第二字段2表示二级标识号，具体为阶段方案配置表；第三字段3表示三级标识号，具体为方案周期；第四字段11表示四级标识号，具体为第11号阶段方案。

4.2.2.10 数据类10 过渡约束信息

数据类10定义为过渡约束信息，详见表9。

表9 过渡约束信息

标识号				状态标记	类型标记
10.过渡约束信息 (constraint)	1.阶段过渡约束配置表 (stageConstraintTable)	1.目标阶段编号(index)	n.对象序号	M(n)	Q
		2.过渡约束值(constraint)	n.对象序号	M(n)	Q、S
	2.相位过渡约束配置表 (phaseConstraintTable)	1.目标相位编号(index)	n.对象序号	O(n)	Q
		2.过渡约束值(constraint)	n.对象序号	O(n)	Q、S

标识号示例：

- a) 阶段3的阶段过渡约束值标识号：10.1.2.3，其中第一字段表示一级标识号，具体为过渡约束；第二字段1表示二级标识号，具体为阶段过渡约束配置表；第三字段2表示三级标识号，具体为阶段过渡约束值；第四字段3表示四级标识号，具体为阶段3。

4.2.2.11 数据类11 日计划信息

数据类11定义为日计划信息，详见表10。

- a) 通信协议规定：最大的日计划数量为128；
- b) 通信协议规定：一天最大划分为48个时段。

表10 日计划信息

标识号				状态标记	类型标记
11. 日计划信息 (dayPlan)	1.实际日计划数量 (maxDayPlan)			M	Q
	2. 日计划配置表 (dayPlanCfgTable)	1.日计划编号(index)	n.对象序号	M(n)	Q、S
		2.日计划所属路口编号 (intersectionNo)	n.对象序号	M(n)	Q、S
		3.时段开始时间链 (beginTimeChain)	n.对象序号	M(n)	Q、S
		4.时段执行阶段方案链 (stagePatternChain)	n.对象序号	M(n)	Q、S
		5.时段运行模式链 (runModeChain)	n.对象序号	M(n)	Q、S
		6.时段动作链1 (actionChain1)	n.对象序号	O(n)	Q、S
		7.时段动作链2 (actionChain2)	n.对象序号	O(n)	Q、S

标识号				状态标记	类型标记
		8.时段动作链3 (actionChain3)	n.对象序号	O(n)	Q、S
		9.时段动作链4 (actionChain4)	n.对象序号	O(n)	Q、S
		10.时段动作链5 (actionChain5)	n.对象序号	O(n)	Q、S
		11.时段动作链6 (actionChain6)	n.对象序号	O(n)	Q、S
		12.时段动作链7 (actionChain7)	n.对象序号	O(n)	Q、S
		13.时段动作链8 (actionChain8)	n.对象序号	O(n)	Q、S

标识号示例：

- a) 实际日计划数量的标识号：，其中第一字段11表示一级标识号，具体为日计划；第二字段1表示二级标识号，具体为实际日计划数量。
- b) 第3号日计划所属路口序号的标识号：11.2.2.3，其中第一字段11表示一级标识号，具体为日计划；第二字段2表示二级标识号，具体为日计划配置表；第三字段2表示三级标识号，具体为日计划所属路口序号；第四字段3表示四级标识号，具体为第3号日计划。

4.2.2.12 数据类12 调度信息

数据类12定义为调度信息，详见表11。

通信协议约定：最大的调度表数量为128。

表11 调度信息

标识号				状态标记	类型标记
12.调度信息 (schedule)	1.实际调度表数量 (maxSchedule)			M	Q
	2.调度配置表 (scheduleCfgTable)	1.调度表编号(index)	n.对象序号	M(n)	Q
		2.调度表优先级(priority)	n.对象序号	M(n)	Q、S
		3.调度表星期值(weekday)	n.对象序号	M(n)	Q、S
		4.调度表月份值(month)	n.对象序号	M(n)	Q、S
		5.调度表日期值(date)	n.对象序号	M(n)	Q、S
		6.调度表的日计划(dayPlan)	n.对象序号	M(n)	Q、S

标识号示例：

- a) 实际调度表数量的标识号：12.1，其中第一字段12表示一级标识号，具体为调度表配置；第二字段1表示二级标识号，具体为实际调度表数量。
- b) 第4号调度表优先级的标识号：12.2.3.4，其中第一字段12表示一级标识号，具体为调度表配置；第二字段2表示二级标识号，具体为调度配置表；第三字段3表示三级标识号，具体为调度优先级；第四字段4表示四级标识号，具体为第4号调度表。

4.2.2.13 数据类13 运行状态

数据类13定义为运行状态，详见表12。

表12 运行状态

标识号				状态标记	类型标记
13.运行状态 (state)	1.设备状态 (unitState)	1.检测器状态(detectorState)		M	Q、T
		2.设备模块状态(moduleState)		O	Q、T
		3.机柜门状态(cabinetDoor)		O	Q、T
		4.电压(voltage)		O	Q
		5.电流(current)		O	Q

标识号				状态标记	类型标记
		6.温度(temp)		O	Q
		7.湿度(humidity)		O	Q
		8.水浸(water)		O	Q、T
		9.烟雾(smoke)		O	Q、T
		10.信号机标准时间(globalTime)		M	Q、S
		11.信号机本地时间(localTime)		M	Q、S
	2.控制状态 (runState)	1.路口序号(intersectionNo)		M	Q、T
		2.路口当前运行模式(curRunMode)		M	Q、T
		3.路口当前阶段方案(curStagePattern)		M	Q、T
		4.路口当前阶段(curStage)		M	Q、T
		5.路口当前阶段时长(curStageTime)		M	Q、T
		6.路口当前环方案(curRingPattern)		O	Q、T
		7.路口当前环相位(curPhase)		O	Q、T
		8.路口当前环相位时长(curPhaseTime)		O	Q、T
		9.路口当前周期长度(curCycle)		M	Q、T
		10.当前周期已运行时间(elapsedTime)		M	Q、T
		11.协调状态信息(coordination)		M	Q、T
		12.当前时间(curTime)		M	Q、T

标识号示例：

- a) 检测器状态的标识号：13.1.1，其中第一字段13表示一级标识号，具体为运行状态；第二字段1表示二级标识号，具体为设备状态；第三字段1表示三级标识号，具体为检测器状态。
- b) 第2个控制路口当前方案的标识号：13.2.3.2，其中第一字段13表示一级标识号，具体为运行状态；第二字段2表示二级标识号，具体为运行状态表；第三字段3表示三级标识号，具体为路口当前方案；第四字段2表示四级标识号，具体为第2控制路口。

4.2.2.14 数据类14 交通数据

数据类14定义为交通数据，详见表13。

表13 交通数据

标识号				状态标记	类型标记
14.交通数据 (trafficData)	1.实时数据 (detectorState)			M	Q
	2.检测器统计数据表 (statisticTable)	1.检测器编号(index)	n.对象序号	M(n)	Q
		2.流量(flow)	n.对象序号	O(n)	Q、T
		3.平均占有率(occupancy)	n.对象序号	O(n)	Q、T
		4.平均车速(avgSpeed)	n.对象序号	O(n)	Q、T
		5.平均车头时距 (avgTimeHeadway)	n.对象序号	O(n)	Q、T
		6.平均车间时距(avgTimeGap)	n.对象序号	O(n)	Q、T
		7.平均停车次数(avgStopTimes)	n.对象序号	O(n)	Q、T
		8.平均排队长度(avgQueneLength)	n.对象序号	O(n)	Q、T

标识号				状态标记	类型标记
		9.最大排队长度(maxQueueLength)	n.对象序号	O(n)	Q、T
		10.等候区平均等候人数(avgPedWaitNum)	n.对象序号	O(n)	Q、T
		11.等候区平均等候时间(avgPedWaitTime)	n.对象序号	O(n)	Q、T
		12.等候区最大等候人数(maxPedWaitNum)	n.对象序号	O(n)	Q、T
		13.等候区最大等候时间(maxPedWaitTime)	n.对象序号	O(n)	Q、T
		14.统计起始时间(startTime)	n.对象序号	O(n)	Q、T
		15.统计结束时间(endTime)	n.对象序号	O(n)	Q、T

标识号示例：

- a) 实时数据的标识号：14.1，其中第一字段14表示一级标识号，具体为交通数据；第二字段1表示二级标识号，具体为实时数据。
- b) 第3检测器流量的标识号：14.2.2.3，其中第一字段14表示一级标识号，具体为交通数据；第二字段2表示二级标识号，具体为统计数据表；第三字段2表示三级标识号，具体为检测器流量；第四字段3表示四级标识号，具体为第3检测器。

4.2.2.15 数据类15 报警数据

数据类15定义为报警数据，详见表14。

表14 报警数据

标识号				状态标记	类型标记
15.报警数据(alarm)	1.当前报警数量(maxAlarm)			M	Q
	2.报警数据表(alarmTable)	1.报警编号(index)	n.对象序号	M(n)	Q、T
		2.报警类型(type)	n.对象序号	O(n)	Q、T
		3.报警值(value)	n.对象序号	O(n)	Q、T
		4.报警时间(time)	n.对象序号	O(n)	Q、T
		5.报警详细信息(detail)	n.对象序号	O(n)	Q、T

标识号示例：

- a) 当前报警数量的标识号：15.1，其中第一字段15表示一级标识号，具体为报警数据；第二字段1表示二级标识号，具体为当前报警数量。
- b) 第1号报警数据值的标识号：15.2.3.1，其中第一字段15表示一级标识号，具体为报警数据；第二字段2表示二级标识号，具体为报警数据表；第三字段3表示三级标识号，具体为报警值；第四字段1表示四级标识号，具体为第1号报警数据。

4.2.2.16 数据类16 故障数据

数据类16定义为故障数据，详见表15。

表15 故障数据

标识号				状态标记	类型标记
16.故障数据(fault)	1.当前故障记录数(maxFault)			M	Q
	2.故障记录表(faultTable)	1.故障编号(index)	n.对象序号	M(n)	Q、T
		2.故障类型(type)	n.对象序号	O(n)	Q、T

标识号				状态标记	类型标记
		3.故障事件发生时间(time)	n.对象序号	O(n)	Q、T
		4.故障动作参数(action)	n.对象序号	O(n)	Q、T
		5.故障详细信息(detail)	n.对象序号	O(n)	Q、T

标识号示例：

- a) 当前故障数量的标识号：16.1，其中第一字段16表示一级标识号，具体为故障数据；第二字段1表示二级标识号，具体为当前故障记录数。
- b) 第1号故障事件发生时间的标识号：16.2.3.1，其中第一字段16表示一级标识号，具体为故障数据；第二字段2表示二级标识号，具体为故障记录表；第三字段3表示三级标识号，具体为故障事件发生时间；第四字段1表示四级标识号，具体为第1号故障。

4.2.2.17 数据类17 中心控制

数据类17定义为中心控制，详见表16。

表16 中心控制

标识号				状态标记	类型标记
17.中心控制 (remoteControl)	1.中心控制命令 (command)	1.指定状态 (assignState)	1.指定阶段(assignStage)	M	Q、S
			2.指定相位(assignPhase)	O	Q、S
			3.指定时长(assignTime)	M	Q、S
			4.指定模式(assignMode)	O	Q、S
		2.指定方案 (assignPattern)	1.指定阶段方案 (assignStagePattern)	M	Q、S
			2.指定环方案 (assignRingPattern)	O	Q、S
			3.指定运行周期数 (assignCycleTimes)	M	Q、S
			4.指定预计生效时间 (assignActionDelay)	M	Q、S
			5.指定模式(assignMode)	M	Q、S
		3.方向锁定配置表 (dirLockCfgTable)	1.方向序号(dirIndex)	M(n)	Q
			2.方向锁定阶段 (dirLockStage)	M(n)	Q、S
			3.方向锁定环相位链 (dirLockPhaseChain)	O(n)	Q、S
	2.协调控制信息 (coordination)	1.协调子区周期(cycle)		O	Q、S
		2.协调相位差(offset)		O	Q、S
		3.相位差调节请求 (offsetAdjRequest)		O	Q、S
		4.调整周期数 (adjCycleTimes)		O	Q、S
		5.协调绿信比模式 (splitMode)		O	Q、S
		6.协调阶段控制 (stageControl)	1.协调阶段 (coordinationStage)	O	Q、S
			2.指定协调阶段方案 (coordinationPattern)	O	Q、S
			3.协调阶段绿信比参数	O	Q、S

标识号				状态标记	类型标记
			(coordinationSplit)		
		7.协调环控制 (ringControl)	1.协调相位 (coordinationPhase)	O	Q、S
			2.指定协调环方案 (coordinationPattern)	O	Q、S
			3.协调环绿信比参数 (coordinationSplit)	O	Q、S
	3.中心临时方案 (tempPattern)	1.方案运行模式 (runMode)		M	Q、S
		2.方案相位差时间 (offset)		M	Q、S
		3.方案相位差计算参考 点(offsetSyncPoint)		M	Q、S
		4.临时方案运行周期数 (totalCycleTimes)		M	Q、S
		5.临时方案执行状态 (runState)		M	Q
		6.临时方案运行剩余周 期数 (remainderCycleTimes)		M	Q
		7.临时阶段方案 (tempStagePattern)	1.继承阶段方案号 (inheritPatternIndex)	M	Q、S
			2.方案协调阶段 (coordinationStage)	M	Q、S
			3.方案阶段时间链(split)	M	Q、S
			4.方案阶段感应链 (detectChain)	O	Q、S
		8.临时环方案 (tempRingPattern)	1.继承环方案号 (inheritPatternIndex)	O	Q、S
			2.方案协调相位 (coordinationPhase)	O	Q、S
			3.方案环绿信比时间链 (splitChain)	O	Q、S
			4.方案环相位感应链 (detectChain)	O	Q、S

标识号示例：

- a) 指定路口2的模式 标识号：17.1.4.2，其中第一字段17表示一级标识号，具体为中心控制；第二字段1表示二级标识号，具体为中心控制命令表；第三字段4表示三级标识号，具体为指定模式；第四字段2表示四级标识号，具体为路口2。

4.2.2.18 数据类18 命令管道

数据类18定义为命令管道，详见表17。

表17 命令管道

标识号			状态标记	类型标记
18.命令管道(pipe)	1.命令值(command)		O	Q、S

标识号示例：

- a) 向信号机发送命令标识号：18.1，其中第一字段18表示一级标识号，具体为命令管道；第二字段1为二级标识号，具体为命令值。

4.2.2.19 数据类19 环方案信息

数据类19定义为环方案信息，详见表18。

当交通信号机采用环语义运行时，本数据类实际生效。当采用阶段语义时，本数据类不生效，任何对本数据类的访问都将返回无访问权限错误。

- a) 通信协议规定：环方案数量最大为128；
b) 通信协议规定：环方案的环数量最大为8，每个环中相位的数量最大为16；

表18 环方案信息

标识号				状态标记	类型标记	
19.环方案信息 (ringPattern)	1.实际方案数 (maxPattern)			O	Q	
	2.方案配置表 (patternCfgTable)	1.方案编号(index)	n.对象序号	O(n)	Q、S	
		2.方案周期(cycle)	n.对象序号	O(n)	Q、S	
		3.方案的协调相位序号 (coordinationIndex)	n.对象序号	O(n)	Q、S	
		4.方案相位差时间(offset)	n.对象序号	O(n)	Q、S	
		5.方案相位差计算参考点 (offsetSyncPoint)	n.对象序号	O(n)	Q、S	
		6.方案环数量(ringNum)	n.对象序号	O(n)	Q、S	
		7.方案环配置表 (ringCfgTable)	1.环的编号 (index)	m.n.对象序号	O(n)	Q、S
			2.环的相位链 (phaseChain)	m.n.对象序号	O(n)	Q、S
			3.环的相位出现类型链 (phaseTypeChain)	m.n.对象序号	O(n)	Q、S
			4.环的绿信比时间链(splitChain)	m.n.对象序号	O(n)	Q、S
			5.环的相位感应链(detectorChain)	m.n.对象序号	O(n)	Q、S
		8.自适应最小周期 (adaptiveMinCycle)	n.对象序号	O(n)	Q、S	
		9.自适应最大周期 (adaptiveMaxCycle)	n.对象序号	O(n)	Q、S	
		10.自适应最大周期波动 (adaptiveMaxCycleDelta)	n.对象序号	O(n)	Q、S	

标识号示例：

- a) 实际环方案数的标识号：19.1，其中第一字段19表示一级标识号，具体为环方案信息；第二字段1表示二级标识号，具体为实际方案数。
- b) 环方案1的周期的标识号：19.2.3.1，其中第一字段19表示一级标识号，具体为环方案信息；第二字段2表示二级标识号，具体为环方案配置表；第三字段3表示三级标识号，具体为环方案周期；第四字段1表示四级标识号，具体为环方案1。
- c) 环方案3的环2相位链的标识号：19.2.7.2.2.3，其中第一字段19表示一级标识号，具体为环方案信息；第二字段2表示二级标识号，具体为环方案配置表；第三字段7表示三级标识号，具体为环方案配置表；第四字段2表示四级标识号，具体为环的相位链；第五字段2表示五级标识号，具体为环的相位链中的第2个元素，即第2个环的相位链；第六字段3表示六级标识号，具体为方案配置表中的第3个元素，即第3个环方案。

4.2.2.20 数据类20 跟随相位信息

数据类20定义为跟随相位信息，详见表19。

当交通信号机采用环语义运行时，本数据类实际生效。当采用阶段语义时，本数据类不生效，任何对本数据类的访问都将返回无访问权限错误。

通信协议规定：信号机最大跟随相位数为64，编号为：1~64。

表19 跟随相位信息

标识号				状态标记	类型标记
20.跟随相位信息 (overlap)	1.实际配置跟随相位数 (maxOverlap)			O	Q
	2.跟随相位配置表 (overlapCfgTable)	1.跟随相位编号(index)	n.对象序号	O(n)	Q
		2.跟随模式(mode)	n.对象序号	O(n)	Q、S
		3.跟随相位的通道 (channelIncluded)	n.对象序号	O(n)	Q、S
		4.跟随的母相位编号 (includedPhase)	n.对象序号	O(n)	Q、S
		5.跟随的修正相位编号 (modifierPhase)	n.对象序号	O(n)	Q、S
		6.跟随相位晚启动时间 (lateBeginTime)	n.对象序号	O(n)	Q、S
		7.跟随相位早结束时间 (earlyEndTime)	n.对象序号	O(n)	Q、S
		8.跟随相位路权延长时间 (trailingTime)	n.对象序号	O(n)	Q、S

标识号示例：

- a) 实际配置的跟随相位数的标识号：20.1，其中第一字段20表示一级标识号，具体为跟随相位信息；第二字段1表示二级标识号，具体为实际配置跟随相位数。
- b) 跟随相位1的母相位编号的标识号：20.2.3.1，其中第一字段20表示一级标识号，具体为跟随相位信息；第二字段2表示二级标识号，具体为跟随相位配置表；第三字段3表示三级标识号，具体为跟随的母相位编号；第四字段1表示四级标识号，具体为跟随相位1。

4.2.2.21 数据类21 通道时段控制计划信息：

数据类21定义为通道时段控制计划信息，详见表20。

表20 通道时段控制计划信息

标识号					状态标记	类型标记
21.通道时段控制 计划信息 (channelTimePlan)	1.实际通道时 段控制计划表 数量(maxPlan)				O	Q
	2.通道时段控 制计划表 (planCfgTable)	1.计划表编号(index)		n.对象序号	O (n)	Q
		2.通道数量(maxChannel)		n.对象序号	O (n)	Q、S
		3.通道时段配 置表 (timeSlotTable)	1.通道编号 (index)	m.n.对象序 号	O (n)	Q
			2.配置标记 (enable)	m.n.对象序 号	O (n)	Q、S
			3.时段数量 (maxTimeSlot)	m.n.对象序 号	O (n)	Q、S
	4.时段开始时间 链 (beginTimeChain)		m.n.对象序 号	O (n)	Q、S	

标识号					状态标记	类型标记
			5.时段结束时间链 (endTimeChain)	m.n.对象序号	O (n)	Q、S
			6.时段控制链 (controlChain)	m.n.对象序号	O (n)	Q、S

标识号示例：

- a) 实际通道时段控制计划表数量的标识号：21.1，其中第一字段21表示一级标识号，具体为通道时段控制计划信息；第二字段1表示二级标识号，具体为实际通道时段控制计划表数量。
- b) 通道时段控制计划表3的通道数量的标识号：21.2.2.3，其中第一字段21表示一级标识号，具体为通道时段控制计划信息；第二字段2表示二级标识号，具体为通道时段控制计划表；第三字段2表示三级标识号，具体为通道数量；第四字段3表示四级标识号，具体为通道时段控制计划表3。
- c) 通道时段控制计划表3的通道7时段配置表的时段数量的标识号：21.2.3.3.7.3，其中第一字段21表示一级标识号，具体为通道时段控制计划信息；第二字段2表示二级标识号，具体为通道时段控制计划表；第三字段3表示三级标识号，具体为通道时段配置表；第四字段3表示四级标识号，具体为时段数量；第五字段7表示五级标识号，具体为通道7；第六字段3为六级标识号，具体为时段控制计划表3。

4.2.2.22 数据类22 通道时段控制调度信息

数据类22定义为通道时段控制调度表信息，详见表21。

表21 通道时段控制调度信息

标识号				状态标记	类型标记
22.通道时段控制调度表信息 (channelTimeCtlShedule)	1.实际通道时段控制调度表数量 (maxShedule)			O	Q
	2.通道时段控制调度表 (sheduleCfgTable)	1.调度表编号(index)	n.对象序号	O(n)	Q
		2.调度优先级(priority)	n.对象序号	O(n)	Q、S
		3.调度表星期值(day)	n.对象序号	O(n)	Q、S
		4.调度表月份值(month)	n.对象序号	O(n)	Q、S
		5.调度表日期值(date)	n.对象序号	O(n)	Q、S
		6.通道时段控制计划(planIndex)	n.对象序号	O(n)	Q、S

标识号示例：

- a) 实际通道时段控制调度表数量的标识号：22.1，其中第一字段22表示一级标识号，具体为通道时段控制调度信息；第二字段1表示二级标识号，具体为实际通道时段控制调度表数量。
- b) 通道时段控制调度表1的调度优先级的标识号：22.2.3.1，其中第一字段22表示一级标识号，具体为通道时段控制调度信息；第二字段2表示二级标识号，具体为通道时段控制调度表；第三字段3表示三级标识号，具体为调度优先级；第四字段1表示四级标识号，具体为通道时段控制调度表1。

4.2.2.23 数据类23 用户信息

数据类23定义为用户组信息，详见表22。

表22 用户信息

标识号				状态标记	类型标记
23.用户信息(user)	1.实际用户数 (maxUser)			O	Q

	2.用户信息表 (userInfoTable)	1.用户编号(index)	n.对象序号	O(n)	Q
		2.用户名称(name)	n.对象序号	O(n)	Q、S
		3.用户密钥(key)	n.对象序号	O(n)	Q、S
		4.用户权限(authority)	n.对象序号	O(n)	Q、S
		5.最近访问时间(accessTime)	n.对象序号	O(n)	Q

标识号示例：

- a) 实际配置的用户数的标识号：23.1，其中第一字段23表示一级标识号，具体为用户信息；第二字段1表示二级标识号，具体为实际配置用户数。
- b) 用户1的名称标识号：23.2.2.1，其中第一字段23表示一级标识号，具体为用户信息；第二字段2表示二级标识号，具体为用户配置表；第三字段2表示三级标识号，具体为用户名称；第四字段1表示四级标识号，具体为用户1。

4.2.3 协议扩展

通信协议约定数据类128（含）以后为厂家数据类扩展，厂家可以根据信号机内部的数据格式自由扩展数据通信协议，通信格式满足协议帧格式即可。一旦某种数据成为所有厂家信号机所共有的数据，则修订通信协议使其成为协议强制数据类。

4.3 数据值MIB

4.3.1 信息数据库MIB HEADER

```
ASC DEFINITIONS ::= BEGIN
-- 下列目标定义用于交通信号控制机设备管理信息数据库
-- the following OBJECT IDENTIFIERS are used in the ASC MIB:
IMPORTS
    Counter
        FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC-1212
asc OBJECT IDENTIFIER ::= { devices 2 }
```

4.3.2 数据类MIB定义

4.3.2.1 数据类1 通用数据

```
generalConfig OBJECT IDENTIFIER ::= { asc 1 }
-- 定义通用数据信息节点
-- 此节点中包含与信号控制机通用数据信息相关的所有数据对象
-- MIB详细定义见总则
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.通用数据(1.3.6.1.4.1.61332.3.2.2.1)。
```

4.3.2.2 数据类2 基础信息

```
basic OBJECT IDENTIFIER ::= { asc 2 }
-- 定义基础信息节点
-- 此节点中包含与信号控制机基础信息相关的所有数据对象
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.基础信息(1.3.6.1.4.1.61332.3.2.2.2)。
```

4.3.2.2.1 信号机编号

```
controllerID OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS           read-write
    STATUS           mandatory
```

DESCRIPTION

“信号机在整个上端系统中唯一编号。整数型。”

::= { basic 1 }

4.3.2.2.2 机柜编码

cabinetID OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS optional

DESCRIPTION

“机柜的硬件编码，一般由固定在机柜上的不可远程更改的硬件电路实现，通常用于校验配置参数是否与本地匹配，整形。”

::= { basic 2 }

4.3.2.2.3 信号机控制路口数量

intersectionNum OBJECT-TYPE

SYNTAX INTEGER(1..8)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该信号机独立控制的实际路口数量。整数型，范围1~8。

协议允许一台信号机控制多个路口，规程详见附录B.3”

::= { basic 3 }

4.3.2.2.4 信号机控制路口属性表

intersectionInfoTable OBJECT-TYPE

SYNTAX SEQUENCE OF intersectionInfoEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“信号机控制多个路口时，一个包含每个路口信息的数据组，数组元素数量等于 intersectionNum。”

::= { basic 4 }

intersectionInfoEntry OBJECT-TYPE

SYNTAX intersectionInfoEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“信号机控制多个路口时，每个路口的信息数据组。这是一个数据序列。”

INDEX { index }

::= { intersectionInfoTable 1 }

intersectionInfoEntry ::= SEQUENCE{

index INTEGER,

globalID OCTET STRING),

comPort INTEGER}

4.3.2.2.3.1 控制路口编号

index OBJECT-TYPE

SYNTAX INTEGER(1..8)

ACCESS read-only

STATUS mandatory

DESCRIPTION

“单台信号机控制多个路口时，信号机内关于对应路口的序号，整数型，路口1的路口编号为1，以此类推。”

::= { intersectionInfoEntry 1 }

4.3.2.2.3.2 控制路口全局编号

globalID OBJECT-TYPE

SYNTAX OCTET STRING(SIZE(0..16))

```

ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "16位数字，全局唯一。
    按照GA/T 2151-2024《道路交通车路协同信息服务通用技术要求》的要求，
    路口编号为13位数字；
    按照GA/T 1049.2-2013《公安交通集成指挥平台通信协议第2部分交通信号
    控制系统》的要求，路口编号为6位行政区划代码+3位区域编号+5位路口编号；
    按照GA/T 1743-2020《道路交通信号控制机信息发布接口规范》的要求，发
    送方标识为7字节，行政区划代码+类型+编号，行政区划代码为3字节，类型为
    2字节，编号为2字节；
    实际制定路口全局编号时建议参考以上规范。"
::= { intersectionInfoEntry 2 }

```

4.3.2.2.3.3 控制路口访问端口

```

comPort OBJECT-TYPE
    SYNTAX          INTERGER
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION
        "当一台信号机控制多个路口时，不同路口所采用的访问端口号配置。
        端口访问规则详见附录B.3。"
    ::= { intersectionInfoEntry 3 }

```

4.3.2.2.4 信号语义

```

semanticCfg OBJECT IDENTIFIER
::= { basic 4 }
-- 定义信号语义信息节点
-- 此节点中包含与信号控制机交通语义相关的所有数据对象

```

4.3.2.2.4.1 支持语义类型

```

supportSemantic OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(2))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "信号机支持的交通语义，Array[0]对应阶段，Array[1]对应环。当支持该
        语义时，对应字节为非零，否则为0。
        为保持协议标准的前向兼容，符合本标准的信号机必须支持阶段语义。"
    ::= { semanticCfg 1 }

```

4.3.2.2.4.2 当前应用语义类型

```

useSemantic OBJECT-TYPE
    SYNTAX          INTEGER(
                        0 (Semantic_PhaseStage),
                        1 (Semantic_PhaseRing))
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION
        "当前采用的应用语义类型。
        0对应阶段语义，1对应环语义，2对应色步语义。
        应用语义在信号机运行状态下不允许改变，任何对此对象的修改只有在信号机
        重启后才会生效。"
    ::= { semanticCfg 2 }

```

4.3.2.2.5 时间操作配置

```

timeCfg OBJECT IDENTIFIER
::= { basic 5 }
-- 定义时间操作配置信息节点

```

-- 此节点中包含与信号控制机时间操作相关的所有数据对象

4.3.2.2.5.1 时间计算基准

```
timeDatum OBJECT-TYPE
    SYNTAX          INTEGER(
                        0 (Time_Ref_Unavailable),
                        1 (Time_Ref_1970),
                        2 (Time_Ref_Everyday),
                        255 (Time_Ref_Undefined))
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "信号机计算协调控制相位差的时间基准，定义如下：
        0：未配置(Time_Ref_Unavailable)；
        1：采用相对1970年1月1日0时刻的绝对相位差(Time_Ref_1970)；
        2：采用相对每天0时刻的相对相位差(Time_Ref_Everyday)；
        255：本标准未涵盖的方式(Time_Ref_Undefined)。"
 ::= { timeCfg 1 }
```

4.3.2.2.5.2 时间获取方式

```
timeGetMethod OBJECT-TYPE
    SYNTAX          INTEGER(
                        0 (Time_Method_Unavailable),
                        1 (Time_Method_RTC),
                        2 (Time_Method_GNSS),
                        3 (Time_Method_NTP),
                        255 (Time_Method_Undefined))
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "信号机时间获取方式，定义如下：
        0：未配置(Time_Method_Unavailable)；
        1：本地实时时钟(Time_Method_RTC)；
        2：卫星定位时钟(Time_Method_GNSS)；
        3：网络时钟(NTP服务)(Time_Method_NTP)；
        255：本标准未涵盖的方式(Time_Method_Undefined)。"
 ::= { timeCfg 2 }
```

4.3.2.2.5.3 卫星时钟标志

```
GNSSClock OBJECT-TYPE
    SYNTAX          INTEGER(
                        0(unavailable),
                        1(available))
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "卫星时钟是否有效，0：无效；1：有效。整数型。"
 ::= { timeCfg 3 }
```

4.3.2.2.6 上位机IPV4配置

```
managerIPv4Cfg OBJECT IDENTIFIER
 ::= { basic 6 }
```

-- 定义上位机IPV4配置信息节点

-- 此节点中包含与信号控制机上位机IPV4相关的所有数据对象

4.3.2.2.6.1 上位机IPV4的IP地址

```
IP OBJECT-TYPE
    SYNTAX          IPAddress
    ACCESS           read-write
    STATUS           optional
```

DESCRIPTION

“上位机的IPv4地址设置。”

```
::= { managerIPv4Cfg 1 }
```

4.3.2.2.6.2 上位机IPv4的通信端口设置

comPort OBJECT-TYPE

SYNTAX INTEGER(0..65535)

ACCESS read-write

STATUS optional

DESCRIPTION

“上位机的IPv4通信端口。”

```
::= { managerIPv4Cfg 2 }
```

4.3.2.2.6.3 上位机IPv4的通信类型设置

comType OBJECT-TYPE

SYNTAX INTEGER(
0 (COM_Type_Unavailable),
1 (COM_Type_TCP),
2 (COM_Type_UDP),
255 (COM_Type_Undefined))

ACCESS read-write

STATUS optional

DESCRIPTION

“上位机的IPv4通信类型设置，定义如下：

0：未定义 (COM_Type_Unavailable)；

1：TCP通信 (COM_Type_TCP)；

2：UDP通信 (COM_Type_UDP)；

255：本标准未涵盖的类型 (COM_Type_Undefined)。”

```
::= { managerIPv4Cfg 3 }
```

4.3.2.2.7 上位机IPv6配置

managerIPv6Cfg OBJECT IDENTIFIER

```
::= { basic 7 }
```

-- 定义上位机IPv6配置信息节点

-- 此节点中包含与信号控制机上位机IPv6相关的所有数据对象

4.3.2.2.7.1 上位机IPv6的IP地址

IP OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(16))

ACCESS read-write

STATUS optional

DESCRIPTION

“上位机的IPv6地址设置，共16字节。Array[0]代表上位机IP的最高地址字节，Array[15]代表上位机IP的最低地址字节。”

```
::= { managerIPv6Cfg 1 }
```

4.3.2.2.7.2 上位机IPv6的通信端口设置

comPort OBJECT-TYPE

SYNTAX INTEGER(0..65535)

ACCESS read-write

STATUS optional

DESCRIPTION

“上位机的IPv6通信端口。”

```
::= { managerIPv6Cfg 2 }
```

4.3.2.2.7.3 上位机IPv6的通信类型设置

comType OBJECT-TYPE

SYNTAX INTERER(
0 (COM_Type_Unavailable),
1 (COM_Type_TCP),
2 (COM_Type_UDP),

```

                255 (COM_Type_Undefined))
ACCESS          read-write
STATUS          optional
DESCRIPTION
    "上位机的IPv6通信类型设置，定义如下：
    0：未定义 (COM_Type_Unavailable)；
    1：TCP通信 (COM_Type_TCP)；
    2：UDP通信 (COM_Type_UDP)；
    255：本标准未涵盖的类型 (COM_Type_Undefined)。"
 ::= { managerIPv6Cfg 3 }

```

4.3.2.3 数据类3 通道信息

```

channel OBJECT IDENTIFIER ::= { asc 3 }
-- 定义通道信息节点
-- 此节点中包含与信号控制机通道信息相关的所有数据对象
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.通道信息 (1.3.6.1.4.1.61332.3.2.2.3)。

```

4.3.2.3.1 实际通道数

```

maxChannel OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "该信号机实际使用的通道编号的最大值，整数型，范围1~64。"
 ::= { channel 1 }

```

4.3.2.3.2 通道配置表

```

channelCfgTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF channelCfgEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION
        "包含每个通道配置的信息数据组，数组元素数量等于maxChannel。"
 ::= { channel 2 }

```

```

channelCfgEntry OBJECT-TYPE
    SYNTAX          channelCfgEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION
        "每个通道的信息数据组。这是一个数据序列。"
    INDEX          { index }
 ::= { channelCfgTable 1 }

```

```

channelCfgEntry ::= SEQUENCE{
    index          INTEGER,
    type           INTEGER,
    intersectionIndex  INTEGER,
    direction      INTEGER,
    movement       INTEGER,
    laneID         SEQUENCE OF INTEGER,
    greenType      INTEGER,
    greenIncluded  SEQUENCE OF INTEGER,
    greenAction    INTEGER,
    V2I_ID         OCTET STRING}

```

4.3.2.3.2.1 通道编号

```

index OBJECT-TYPE

```

SYNTAX INTEGER(1..64)
ACCESS read-only
STATUS mandatory
DESCRIPTION

“信号机内关于通道的序号，整数型，通道1的编号为1，以此类推，最大64个通道。”

::= { channelCfgEntry 1 }

4.3.2.3.2.2 通道灯组类型

type OBJECT-TYPE

SYNTAX INTEGER(1..64)
ACCESS read-write
STATUS mandatory
DESCRIPTION

“灯组类型定义为一个32位无符号数，其中第一字节（Bit0～Bit7）代表灯组功能，第二字节（Bit8～Bit15）代表灯组形状，第三字节（Bit16～Bit23）代表多功能灯的组合模式，第四字节（Bit24～Bit31）代表多功能灯的关联通道编号。

第一字节（Bit0～Bit7），主要定义灯组服务对象，详细定义如下：

- a) 未配置 (Light_Type_Unavailable)
0x00
- b) 机动车灯组 (Light_Type_Vehicle)
0x01
- c) 非机动车灯组 (Light_Type_NonVehicle)
0x02
- d) 行人灯组 (Light_Type_Pedestrian)
0x03
- e) 车道灯组 (Light_Type_Road)
0x04
- f) 可变交通标志 (Light_Type_VMS)
0x05
- g) 公交专用灯组 (Light_Type_Bus)
0x06
- h) 有轨电车专用灯组 (Light_Type_Tramcar)
0x07
- i) 特殊灯组 (Light_Type_Special)
0x08
- j) 本标准未涵盖的灯组类型 (Light_Type_Undefined)
0xFF

第二字节（Bit8～Bit15），定义灯组形状。

该字节定义分为低4位和高4位，其中低4位（Bit8～Bit11）定义灯组的默认显示形状，高4位（Bit12～Bit15）定义当灯组为多功能灯组时，显示的第二种形状。

针对每个4位数据，灯组形状详细定义如下：

- a) 未配置 (Light_Shape_Unavailable)
0x00
- b) 圆盘灯 (Light_Shape_Ball)
0x01
- c) 直行箭头灯 (Light_Shape_Straight)
0x02
- d) 左转箭头灯 (Light_Shape_Left)
0x03
- e) 右转箭头灯 (Light_Shape_Right)
0x04
- f) 掉头箭头灯 (Light_Shape_UTurn)
0x05

- g) 左转加直行箭头灯 (Light_Shape_StraightLeft)
0x06
- h) 右转加直行箭头灯 (Light_Shape_StraightRight)
0x07
- i) 左转加直行加右转箭头灯 (Light_Shape_LeftStraightRight)
0x08
- j) 本标准未涵盖的灯组显示类型 (Light_Shape_Undefined)
0x0F

注1: 根据GB14886-2016规范，左转加直行、右转加直行、左转加直行加右转这几类显示形式不符合规范要求，本标准仍然保持类型定义是为了涵盖市面上已知类型，但并不推荐实际项目中使用。

注2: 本数据主要目的是区分保护式放行和许可式放行在灯具上的差异，针对行人、公交车、非机动车这类有特定图案规范但是目前不涉及两种路权切换的灯具，可以直接采用未配置选项，如果后期发现也存在类似需求再考虑增补定义。

示例: 非多功能灯组，例如单一圆盘灯，本字节配置为0x01；多功能灯组，左转箭头与圆盘灯切换，默认圆盘显示，本字节配置为0x31；左转箭头与圆盘灯切换，默认箭头显示，本字节配置为0x13。

第三字节 (Bit16~Bit23)，定义多功能灯组合模式。

针对多功能灯，通常需要使用信号机的两个通道来实现灯组形状的切换，本类型定义了几种常见的组合模式，详细定义如下：

- a) 未配置 (Light_Channel_Unavailable)
0x00
- b) 主通道 (Light_Channel_Main)
0x01
- c) 从通道模式1 (Light_Channel_SlaveMode1)
0x02
- d) 从通道模式2 (Light_Channel_SlaveMode2)
0x03
- e) 本标准未涵盖的组合类型 (Light_Channel_Undefined)
0xFF

以上几种组合模式的详细解释如下：

主通道+从通道模式1：

此种方式灯组需要接入完整的两个信号机通道的红黄绿灯色驱动，主通道对应默认显示，从通道对应第二种形状显示。当主从通道为互斥关系，即一个通道为放行状态时，关联的另一个通道保持灭灯，则直接显示各自灯形状。当发生两个通道灯色同亮，显示主通道灯形状。

主通道+从通道模式2：

此种方式灯组需要接入完整的两个信号机通道的红黄绿灯色驱动，主通道对应默认显示，从通道对应第二种形状显示。当主从通道为互斥关系，即一个通道为放行状态时，关联的另一个通道保持灭灯，则直接显示各自灯形状。当发生两个通道绿灯同亮，应当显示主通道绿灯形状，当发生两个通道红灯或者黄灯同亮，则应当显示从通道红黄灯形状。

第四字节 (Bit24~Bit31)，定义多功能灯关联通道编号。

针对多功能灯，通常需要使用信号机的两个通道来实现灯组形状的切换，本类型定义了关联的信号机通道。如果本通道为主通道，则关联的信号灯通道为从通道编号；如果本通道为从通道，则关联的信号灯通道为主通道。关联通道编号数据范围1~64。”

::= { channelCfgEntry 2 }

4.3.2.3.2.3 通道归属路口

intersectionIndex OBJECT-TYPE
SYNTAX INTEGER (1..8)
ACCESS read-write
STATUS mandatory
DESCRIPTION

“信号机控制多个路口时，通道对应的路口编号，路口1的编号为1，以此类推。由于路口编号并不在协议层面体现，且通道资源为多路口统一分配，因此信号机配置工具应当具备错误分析能力，防止出现路口和通道混乱。”

```
::= { channelCfgEntry 3 }
```

4.3.2.3.2.4 通道对应交通流的方向

direction OBJECT-TYPE

```
SYNTAX          INTEGER(
                    0(Unavailable),
                    1(East),
                    2(West),
                    3(South),
                    4(North),
                    5(Northeast),
                    6(Northwest),
                    7(Southeast),
                    8(Southwest),
                    255(Undefined))
```

ACCESS read-write

STATUS optional

DESCRIPTION

“通道对应的交通流的方向（来向），定义如下：

a) 未配置 (Unavailable)	0x00
b) 东方向 (East)	0x01
c) 西方向 (West)	0x02
d) 南方向 (South)	0x03
e) 北方向 (North)	0x04
f) 东北方向 (Northeast)	0x05
g) 西北方向 (Northwest)	0x06
h) 东南方向 (Southeast)	0x07
i) 西南方向 (Southwest)	0x08
j) 本标准未涵盖的类型 (Undefined)	0xFF”

```
::= { channelCfgEntry 4 }
```

4.3.2.3.2.5 通道对应交通流的流向

movement OBJECT-TYPE

```
SYNTAX          INTEGER(
                    0(Unavailable),
                    1(Left),
                    2(Straight),
                    3(Right),
                    4(Straight_Left),
                    5(Straight_Right),
                    6(Left_Right),
                    7(Left_Straight_Right),
                    8(UTurn),
                    9(UTurn_Left),
                    10(UTurn_Straight),
                    11(UTurn_Right),
                    12(UTurn_Straight_Left),
                    13(UTurn_Straight_Right),
                    14(UTurn_Left_Right),
                    15(UTurn_Left_Straight_Right),
                    16(Pedestrian_Cross),
                    17(Pedestrian_Twice_Cross_1),
                    18(Pedestrian_Twice_Cross_2),
                    19(Pedestrian_Waiting_Area_1),
                    20(Pedestrian_Waiting_Area_2),
                    21(Channeling_Pedestrian),
                    22(Relief_Road_Pedestrian_Cross_1),
                    23(Relief_Road_Pedestrian_Cross_2),
                    255(Undefined))
```

ACCESS read-write

STATUS	optional
DESCRIPTION	“通道对应的交通流的流向（去向），定义如下：
a) 未配置 (Unavailable)	0x00
b) 左转 (Left)	0x01
c) 直行 (Straight)	0x02
d) 右转 (Right)	0x03
e) 直左 (Straight_Left)	0x04
f) 直右 (Straight_Right)	0x05
g) 左右 (Left_Right)	0x06
h) 左直右 (Left_Straight_Right)	0x07
i) 掉头 (UTurn)	0x08
j) 掉头左转 (UTurn_Left)	0x09
k) 掉头直行 (UTurn_Straight)	0x0A
l) 掉头右转 (UTurn_Right)	0x0B
m) 掉头直左 (UTurn_Straight_Left)	0x0C
n) 掉头直右 (UTurn_Straight_Right)	0x0D
o) 掉头左右 (UTurn_Left_Right)	0x0E
p) 掉头左直右 (UTurn_Left_Straight_Right)	0x0F
q) 行人一段过街 (Pedestrian_Cross)	0x10
r) 行人一次过街 (Pedestrian_Twice_Cross_1)	0x11
s) 行人二次过街 (Pedestrian_Twice_Cross_2)	0x12
t) 行人等待区1 (Pedestrian_Waiting_Area_1)	0x13
u) 行人等待区2 (Pedestrian_Waiting_Area_2)	0x14
v) 渠化行人 (Channeling_Pedestrian)	0x15
w) 辅道行人一 (Relief_Road_Pedestrian_Cross_1)	0x16
x) 辅道行人二 (Relief_Road_Pedestrian_Cross_2)	0x17
y) 本标准未涵盖的类型 (Undefined)	0xFF”
::= { channelCfgEntry 5 }	

4.3.2.3.2.6 通道对应的车道

laneID OBJECT-TYPE	
SYNTAX	SEQUENCE OF INTEGER(SIZE(0..16))
ACCESS	read-write
STATUS	optional
DESCRIPTION	“通道对应的车道，每个通道最大对应16个车道。 车道的编号从1开始编号，最大128，0代表无效的车道， 车道的编号规则建议采用方向+序号的方式编码。 方向编码参考“通道对应交通流的方向(4.3.2.)3.2.4(本文后续当出现标识号引用时，将采用此种表示方式，前方小括号内为文章的目录路径，括号以外，为实际标识号)”。 序号采用驾驶人视角编号，最左侧为1车道，例如南向入口左转车道编号为：31（十进制）。 通道对应的车道定义的目的是实现同向车流分车道控制时候的区分。 对于多功能灯组，主从通道都要配置对应的车道。 当采用RAW格式时，长度为16个字节(16 Byte)，每个字节代表所包含的一个车道的编号，不使用的相位的车道字节填入0。 当采用JSON格式时，是一个包含0到16个数字的数组，每个数字代表一个对应车道的编号。”
::= { channelCfgEntry 6 }	

4.3.2.3.2.7 交通流放行规则

greenType OBJECT-TYPE	
SYNTAX	INTEGER(

```

0 (Movement_Type_Unavailable),
1 (Movement_Type_Protected),
2 (Movement_Type_Permissive),
3 (Movement_Type_FlashYellow),
4 (Movement_Type_FlashRed),
255 (Movement_Type_Undefined))
ACCESS      read-write
STATUS      optional
DESCRIPTION

```

“信号机内通道对应交通流放行规则类型值，当通道输出为绿色时，此对象定义该通道的交通流放行规则种类，该数据用于生成SAE J2735标准的SPaT数据，详细值定义如下：

- a) 未配置 (Movement_Type_Unavailable) 0x00
- b) 保护式放行 (Movement_Type_Protected) 0x01
表示绿灯期间至少有一段时间处在保护式通行模式下，这段时间具有优先通行权。注意，路权有可能在保护式通行和许可式通行之间切换，只要存在保护式通行的区间，类型定义就是保护式放行。
- c) 许可式放行 (Movement_Type_Permissive) 0x02
表示只有不和行人和/或任何交通流冲突的状况下，才允许进行转弯。
- d) 黄闪 (Movement_Type_FlashYellow) 0x03
表示机动车可以在确保不和行人和/或任何交通流冲突的状况下谨慎行进。
- e) 红闪 (Movement_Type_FlashRed) 0x04
表示机动车应当首先停车，然后可以在确保不和行人和/或任何交通流冲突的状况下行进。
- f) 本标准未定义的放行规则种类 (Movement_Type_Undefined) 0xFF”

```
 ::= { channelCfgEntry 7 }
```

4.3.2.3.2.8 许可放行关联车道

```
greenIncluded OBJECT-TYPE
```

```

SYNTAX      SEQUENCE OF INTEGER(SIZE(0..64))
ACCESS      read-write
STATUS      optional
DESCRIPTION

```

“当本通道放行规则设定为保护式放行且当前通道状态为绿灯，而许可放行关联通道内有通道不是红灯或者处在灭灯状态，则通道放行许可状态应当为“许可式放行”，否则通道放行许可状态为“保护式放行”。

如果本通道放行规则不是保护式放行，则本属性参数设置忽略。

当采用RAW格式时：

该属性长度为8个字节(64 bit)，bit0代表信号通道1，bit63代表信号通道64，数据结构为BYTE8、...、BYTE1，代表 bit63~bit56、...、bit7~bit0，相应的 bit位为1 时，表示该通道与当前通道交通流许可式放行规则关联。

当采用JSON格式时：

该属性为包含0到64个数字的数组，每个数字对应一个关联通道编号。”

```
 ::= { channelCfgEntry 8 }
```

4.3.2.3.2.9 保护放行切换许可放行灯组动作

```
greenAction OBJECT-TYPE
```

```

SYNTAX      INTEGER(
0x01 (Light_Status_Off),
0x20 (Light_Status_Green),
0x31 (Light_Status_YellowFlash))
ACCESS      read-write
STATUS      optional
DESCRIPTION

```

“当发生通道从保护式放行切换到许可式放行时，灯组的动作，取值范围参考附录A.3灯组状态。”

```
::= { channelCfgEntry 9 }
```

4.3.2.3.2.10 通道灯组ID

V2I_ID OBJECT-TYPE

SYNTAX OCTET STRING(SIZE(0..255))

ACCESS read-write

STATUS optional

DESCRIPTION

“车路协同广播使用的车道ID。

为实现对SAE J2735标准的支持，该对象用于支持交通信号控制数据的交换，并包含与信号通道输出相关联的交叉口通道的（区域）唯一标识符。”

```
::= { channelCfgEntry 10 }
```

4.3.2.3.3 通道控制表

channelCtlTable OBJECT-TYPE

SYNTAX SEQUENCE OF channelCtlEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“包含每个通道控制的信息数据组，数组元素数量等于maxChannel。”

```
::= { channel 3 }
```

channelCtlEntry OBJECT-TYPE

SYNTAX channelCtlEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“每个通道的控制数据组。这是一个数据序列。”

INDEX { index }

```
::= { channelCtlTable 1 }
```

channelCtlEntry ::= SEQUENCE{

index INTEGER,

mask INTEGER,

Disable INTEGER}

4.3.2.3.3.1 通道编号

index OBJECT-TYPE

SYNTAX INTEGER(1..64)

ACCESS read-only

STATUS mandatory

DESCRIPTION

“信号机内关于通道的序号，整数型，通道1的编号为1，以此类推，最大64个通道。”

```
::= { channelCtlEntry 1 }
```

4.3.2.3.3.2 通道屏蔽

mask OBJECT-TYPE

SYNTAX INTEGER(

0 (Mask_OFF),

1 (Mask_ON))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“信号机内通道屏蔽控制。

0：表示该通道正常输出；

1：表示屏蔽该通道输出（该通道强制变成红灯）”

```
::= { channelCtlEntry 2 }
```

4.3.2.3.3.3 通道禁止

disable OBJECT-TYPE

```

SYNTAX          INTEGER(
                  0(Disable_OFF),
                  1(Disable_ON))
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "信号机内通道禁止控制。
    0：表示该通道正常输出；
    1：表示禁止该通道输出（该通道强制变成灭灯）。
    通道禁止的优先级高于通道屏蔽。"
::= { channelCtlEntry 3 }

```

4.3.2.3.4 通道状态表

```

channelStateTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF channelStateEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION
        "包含每个通道状态的信息数据组，数组元素数量等于maxChannel。"
    ::= { channel 4 }

channelStateEntry OBJECT-TYPE
    SYNTAX          channelStateEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION
        "每个通道的状态数据组。这是一个数据序列。"
    INDEX          { index }
    ::= { channelStateTable 1 }

channelStateEntry ::= SEQUENCE{
    index          INTEGER,
    signalState    INTEGER,
    movementState  INTEGER}

```

4.3.2.3.4.1 通道编号

```

index OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "信号机内关于通道的序号，整数型，通道1的编号为1，以此类推，最大64个通道。"
    ::= { channelStateEntry 1 }

```

4.3.2.3.4.2 通道灯组状态

```

signalState OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION
        "信号机内通道的灯组状态值，详细定义参见附录A.3。"
    ::= { channelStateEntry 2 }

```

4.3.2.3.4.3 交通流放行状态

```

movementState OBJECT-TYPE
    SYNTAX          INTEGER(
        0(Movement_State_Unavailable),
        1(Movement_State_Dark),
        2(Movement_State_StopThenProceed),
        3(Movement_State_StopAndRemain),

```

	4 (Movement_State_PreMovement),
	5 (Movement_State_PermissiveAllowed),
	6 (Movement_State_ProtectedAllowed),
	7 (Movement_State_PermissiveClearance),
	8 (Movement_State_ProtectedClearance),
	9 (Movement_State_CautionConflictingTraffic),
	255 (Movement_State_Undefined))
ACCESS	read-write
STATUS	optional
DESCRIPTION	

“信号机内通道对应交通流当前放行状态，该数据用于生成SAE J2735标准的SPaT数据，详细值定义如下：

- a) 未设置 (Movement_State_Unavailable)
0x00
- b) 不受信号灯控制 (Movement_State_Dark)
0x01
- c) 停车然后启动 (Movement_State_StopThenProceed)
0x02
车辆应当在停车线前停止，然后在确认安全以后行进。
- d) 停车并且等待 (Movement_State_StopAndRemain)
0x03
车辆应当在停车线前停止并保持停车等待状态。
- e) 准备启动 (Movement_State_PreMovement)
0x04
车辆应当停止并做好行进准备。
- f) 许可式通行 (Movement_State_PermissiveAllowed)
0x05
车辆应当在确保不和行人和/或任何交通流冲突的状况下行进。
- g) 保护式通行 (Movement_State_ProtectedAllowed)
0x06
车辆具有优先通行权。
- h) 许可间隙 (Movement_State_PermissiveClearance)
0x07
车辆应当准备停止，如果不能停止可以继续行进，但要注意此时交叉口可能存在冲突交通流。一般此状态跟随许可式通行 (Movement_State_PermissiveAllowed) 发生。
- i) 保护间隙 (Movement_State_ProtectedClearance)
0x08
车辆应当准备停止，如果不能停止可以继续按指示方向行进。一般此状态跟随保护式通行 (Movement_State_ProtectedAllowed) 发生。
- j) 注意冲突交通流 (Movement_State_CautionConflictingTraffic)
0x09
车辆应当在确保安全的前提下行进，注意此时交叉口可能存在冲突交通流。
- k) 本标准未定义的放行规则状态种类 (Movement_State_Undefined)
0xFF

当发生通道对应交通流放行状态变化时，信号机应当主动推送状态变化信息。

对于驱动多功能灯组的通道的变化产生的交通流放行状态变化，由主通道负责推送状态变化。”

```
::= { channelStateEntry 3 }
```

4.3.2.4 数据类4 相位信息

```
phase OBJECT IDENTIFIER
::= { asc 4 }
-- 定义相位信息节点
-- 此节点中包含与信号控制机相位信息相关的所有数据对象
```

- 当使用SNMP协议访问本数据类时，设备标识号路径为：
 iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
 设备.交通信号控制机.相位信息(1.3.6.1.4.1.61332.3.2.2.4)。

4.3.2.4.1 实际相位数

```
maxPhases OBJECT-TYPE
    SYNTAX          INTEGER(2..64)
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "该信号机实际使用的相位编号的最大值，整数型，范围1~64。"
 ::= { phase 1 }
```

4.3.2.4.2 相位配置表

```
phaseCfgTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF phaseCfgEntry
    ACCESS           not-accessible
    STATUS           mandatory
    DESCRIPTION
        "包含每个相位配置的信息数据组，数组元素数量等于maxPhases。"
 ::= { phase 2 }
```

```
phaseCfgEntry OBJECT-TYPE
    SYNTAX          phaseCfgEntry
    ACCESS           not-accessible
    STATUS           mandatory
    DESCRIPTION
        "每个相位的信息数据组。这是一个数据序列。"
    INDEX           { index }
 ::= { phaseCfgTable 1 }
```

```
phaseCfgEntry ::= SEQUENCE{
    index          INTEGER,
    channelIncluded SEQUENCE OF INTEGER,
    onLoseStep1LightType INTEGER,
    onLoseStep1Time    INTEGER,
    onLoseStep2LightType INTEGER,
    onLoseStep2Time    INTEGER,
    onLoseStep3LightType INTEGER,
    onLoseStep3Time    INTEGER,
    onGetStep1LightType INTEGER,
    onGetStep1Time     INTEGER,
    onGetStep2LightType INTEGER,
    onGetStep2Time     INTEGER,
    onGetStep2LightType INTEGER,
    onGetStep2Time     INTEGER,
    onStartupGetStep1LightType INTEGER,
    onStartupGetStep1Time    INTEGER,
    onStartupGetStep2LightType INTEGER,
    onStartupGetStep2Time    INTEGER,
    onStartupGetStep3LightType INTEGER,
    onStartupGetStep3Time    INTEGER,
    onStartupLoseStep1LightType INTEGER,
    onStartupLoseStep1Time    INTEGER,
    onStartupLoseStep2LightType INTEGER,
    onStartupLoseStep2Time    INTEGER,
    onStartupLoseStep3LightType INTEGER,
    onStartupLoseStep3Time    INTEGER,
    minGreen          INTEGER,
    maxGreen1          INTEGER,
    maxGreen2          INTEGER,
```

```

extendedGreen    INTEGER,
phaseRequest     SEQUENCE OF INTEGER,
maxWaste         INTEGER,
lossTimeOnSwitch INTEGER}

```

4.3.2.4.2.1 相位编号

index OBJECT-TYPE

```

SYNTAX          INTEGER(1..64)
ACCESS          read-only
STATUS          mandatory
DESCRIPTION

```

“信号机内关于相位的序号，整数型，相位1的编号为1，以此类推，最大64个相位。”

```
::= { phaseCfgEntry 1 }
```

4.3.2.4.2.2 相位的通道

channelIncluded OBJECT-TYPE

```

SYNTAX          SEQUENCE OF INTEGER(SIZE(0..64))
ACCESS          read-write
STATUS          mandatory
DESCRIPTION

```

“当采用RAW格式时，相位的通道的长度为8个字节(64 bit)，bit0代表信号通道1，bit63代表信号通道64，数据结构为BYTE8、...、BYTE1，代表bit63~bit56、...、bit7~bit0，相应的 bit位为1 时，表示该相位会从该通道上输出。

当采用JSON格式时，相位的通道是一个包含0~64个数字的数组，每个数字代表该相位包含的一个通道的编号，当相位不包含实际的通道时，该字符串长度为0。

注意：相位可以不包含任何通道，此时相位属于虚相位，当使用环逻辑时，可通过使用跟随逻辑关联实际通道。虚相位可以配置对应的相位的需求、相位的车道等属性。”

```
::= { phaseCfgEntry 2 }
```

4.3.2.4.2.3 关闭路权过渡灯色1类型

onLoseStep1LightType OBJECT-TYPE

```

SYNTAX          INTEGER(0..255)
ACCESS          read-write
STATUS          mandatory
DESCRIPTION

```

“该相位失去路权时第一步的过渡灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 3 }
```

4.3.2.4.2.4 关闭路权过渡灯色1时间

onLoseStep1Time OBJECT-TYPE

```

SYNTAX          INTEGER(0..255)
ACCESS          read-write
STATUS          mandatory
DESCRIPTION

```

“该相位失去路权时第一步的过渡时间，单位：1s，为0时，表示该灯色不执行。”

```
::= { phaseCfgEntry 4 }
```

4.3.2.4.2.5 关闭路权过渡灯色2类型

onLoseStep2LightType OBJECT-TYPE

```

SYNTAX          INTEGER(0..255)
ACCESS          read-write
STATUS          mandatory
DESCRIPTION

```

“该相位失去路权时第二步的过渡灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 5 }
```

4.3.2.4.2.6 关闭路权过渡灯色2时间

```
onLoseStep2Time OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "该相位失去路权时第二步的过渡时间，单位：1s，为0时，表示该灯色不执行。
        "
 ::= { phaseCfgEntry 6 }
```

4.3.2.4.2.7 关闭路权过渡灯色3类型

```
onLoseStep3LightType OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "该相位失去路权时第三步的过渡灯态，取值范围详见附录A.3灯组状态。"
 ::= { phaseCfgEntry 7 }
```

4.3.2.4.2.8 关闭路权过渡灯色1时间

```
onLoseStep3Time OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "该相位失去路权时第三步的过渡时间，单位：1s，为0时，表示该灯色不执行。
        "
 ::= { phaseCfgEntry 8 }
```

4.3.2.4.2.9 启动路权过渡灯色1类型

```
onGetStep1LightType OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "该相位获得路权时第一步的过渡灯态，取值范围详见附录A.3灯组状态。"
 ::= { phaseCfgEntry 9 }
```

4.3.2.4.2.10 启动路权过渡灯色1时间

```
onGetStep1Time OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "该相位获得路权时第一步的过渡时间，单位：1s，为0时，表示该灯色不执行。
        "
 ::= { phaseCfgEntry 10 }
```

4.3.2.4.2.11 启动路权过渡灯色2类型

```
onGetStep2LightType OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "该相位获得路权时第二步的过渡灯态，取值范围详见附录A.3灯组状态。"
 ::= { phaseCfgEntry 11 }
```

4.3.2.4.2.12 启动路权过渡灯色2时间

```
onGetStep2Time OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS           read-write
    STATUS           mandatory
```

DESCRIPTION

“该相位获得路权时第二步的过渡时间，单位：1s，为0时，表示该灯色不执行。”

”

```
::= { phaseCfgEntry 12 }
```

4.3.2.4.2.13 启动路权过渡灯色3类型

onGetStep3LightType OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位获得路权时第三步的过渡灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 13 }
```

4.3.2.4.2.14 启动路权过渡灯色1时间

onGetStep3Time OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位获得路权时第三步的过渡时间，单位：1s，为0时，表示该灯色不执行。”

”

```
::= { phaseCfgEntry 14 }
```

4.3.2.4.2.15 开机启动路权灯色1类型

onStartupGetStep1LightType OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果获得路权时第一步的灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 15 }
```

4.3.2.4.2.16 开机启动路权灯色1时间

onStartupGetStep1Time OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果获得路权第一步的时间，单位：1s，为0时，表示该灯色不执行。”

```
::= { phaseCfgEntry 16 }
```

4.3.2.4.2.17 开机启动路权灯色2类型

onStartupGetStep2LightType OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果获得路权时第二步的灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 17 }
```

4.3.2.4.2.18 开机启动路权灯色2时间

onStartupGetStep2Time OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果获得路权第二步的时间，单位：1s，为0时，表示该灯色不执行。”

```
::= { phaseCfgEntry 18 }
```

4.3.2.4.2.19 开机启动路权灯色3类型

onStartupGetStep3LightType OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果获得路权时第三步的灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 19 }
```

4.3.2.4.2.20 开机启动路权灯色3时间

onStartupGetStep3Time OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果获得路权第三步的时间，单位：1s，为0时，表示该灯色不执行。”

```
::= { phaseCfgEntry 20 }
```

4.3.2.4.2.21 开机关闭路权灯色1类型

onStartupLoseStep1LightType OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果关闭路权时第一步的灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 21 }
```

4.3.2.4.2.22 开机关闭路权灯色1时间

onStartupLoseStep1Time OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果关闭路权第一步的时间，单位：1s，为0时，表示该灯色不执行。”

```
::= { phaseCfgEntry 22 }
```

4.3.2.4.2.23 开机关闭路权灯色2类型

onStartupLoseStep2LightType OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果关闭路权时第二步的灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 23 }
```

4.3.2.4.2.24 开机关闭路权灯色2时间

onStartupLoseStep2Time OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果关闭路权第二步的时间，单位：1s，为0时，表示该灯色不执行。”

```
::= { phaseCfgEntry 24 }
```

4.3.2.4.2.25 开机关闭路权灯色3类型

onStartupLoseStep3LightType OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果关闭路权时第三步的灯态，取值范围详见附录A.3灯组状态。”

```
::= { phaseCfgEntry 25 }
```

4.3.2.4.2.26 开机关闭路权灯色3时间

onStartupLoseStep3Time OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位开机时，如果关闭路权第三步的时间，单位：1s，为0时，表示该灯色不执行。”

```
::= { phaseCfgEntry 26 }
```

4.3.2.4.2.27 相位最小绿时间

minGreen OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“该相位必须开启的最短时间，单位：1s。

最短时间的判据目前一般两种考虑：

1. 考虑机动车或者行人或者非机动车安全通过所需要的最小时间；
2. 考虑清空车辆检测器到停止线之间的车辆所需要的时间。”

```
::= { phaseCfgEntry 27 }
```

4.3.2.4.2.28 相位最大绿时间1

maxGreen1 OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“感应控制方式下，当冲突相位存在交通流通行请求时，当前相位绿灯可继续保持的最长时间。单位：1s。”

```
::= { phaseCfgEntry 28 }
```

4.3.2.4.2.29 相位最大绿时间2

maxGreen2 OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“感应控制方式下，当冲突相位存在交通流通行请求时，当前相位绿灯可继续保持的最长时间。单位：1s。”

```
::= { phaseCfgEntry 29 }
```

4.3.2.4.2.30 相位延长绿时间

extendedGreen OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS mandatory
DESCRIPTION

“信号机处于感应控制模式时，该相位延长绿灯的步长，单位：0.1s。”

::= { phaseCfgEntry 30 }

4.3.2.4.2.31 相位的需求

phaseRequest OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..128))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“当采用RAW格式时，相位的需求的长度为16个字节(128 bit)，bit0代表截面型检测器1，bit127代表截面型检测器128。数据结构为BYTE16、...、BYTE1，代表bit127~bit120、...、bit7~bit0，相应的bit位为1时，表示该截面型检测器上存在车辆会请求该相位。

当采用JSON格式时，相位的需求是一个0到128个数字的数组，数组中每个数字代表一个对应的截面型检测器编号，当该相位无对应的截面型检测器时，该数组长度为0。

该属性定义是为了保持和传统习惯兼容，本质上此属性定义将相位的需求视为基于传统的线圈类检测器的检测数据输出，传统信号控制逻辑基于此检测数据产生典型的感应控制逻辑。

目前存在基于雷达、视频类传感器提供的大范围车辆轨迹数据计算最佳放行策略的算法，相位的需求已经不能简单理解为类似线圈的0、1数据，同时新型检测器和相位的对应关系也不能简单地用本数据属性实现对应。

因此本协议增加了相位的车道数据属性，同时检测器也包含了车道编号属性，通过车道实现相位和检测器的关联关系。

具体如何通过检测器的数据产生控制方案本协议不作规定。”

::= { phaseCfgEntry 31 }

4.3.2.4.2.32 相位的最大浪费时间

maxWaste OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

“信号机处于感应控制模式时，该相位的最大浪费时间，单位：1s。

代表相位结束前，最大允许的车辆请求时间间隔。”

::= { phaseCfgEntry 32 }

4.3.2.4.2.33 相位的切换损失时间

lossTimeOnSwitch OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

“该相位切换导致的损失时间，包括最大的浪费时间，单位：0.1s。

该时间包含绿灯初期的启动损失（绿灯刚刚开启，车辆需要从静止开始启动会导致一定的时间损失）以及黄灯末期的损失时间（黄灯即将结束时，行驶的车辆为避免闯红灯提前停车导致的时间损失），用于自适应或者其他智能算法的计算参数。当不使用这类控制算法时，该参数可忽略。”

::= { phaseCfgEntry 33 }

4.3.2.4.2.34 开机黄闪时长

startupFlashYellowTime OBJECT-TYPE

SYNTAX INTEGER(0..300) -- 单位：秒

ACCESS read-write

STATUS mandatory

DESCRIPTION

“信号机开机时进入黄闪状态的持续时间，单位为秒。

该时长用于提示设备上电初始化中，符合《GB 14886-2016》第7.4条关于开机控制要求。

值为0时表示不开启黄闪阶段。"
 ::= { phaseCfgEntry 34 }

4.3.2.4.2.35 开机全红时长

startupAllRedTime OBJECT-TYPE

SYNTAX	INTEGER(0..300)	-- 单位: 秒
ACCESS	read-write	
STATUS	mandatory	
DESCRIPTION		

"信号机开机黄闪结束后进入全红阶段的持续时间, 单位为秒。

该阶段所有灯组为红灯, 用于缓冲系统同步与运行安全, 满足国标对开机安全的要求。

值为0时表示不开启全红阶段。"

::= { phaseCfgEntry 35 }

4.3.2.4.2.36 行人绿灯时间

pedestrianGreenTime OBJECT-TYPE

SYNTAX	INTEGER(0..300)
ACCESS	read-write
STATUS	optional
DESCRIPTION	

"当前相位对应的行人绿灯时间, 单位为秒。

- 表示允许行人通行的有效时间;
- 通常在机动车红灯对应期间内点亮行人绿灯;
- 可配合 minGreen、maxGreen 做协调计算;
- 值为 0 表示该相位不涉及行人通行。"

::= { phaseCfgEntry 36 }

4.3.2.4.2.37 行人清空时间

pedestrianClearTime OBJECT-TYPE

SYNTAX	INTEGER(0..60)
ACCESS	read-write
STATUS	optional
DESCRIPTION	

"当前相位对应的行人清空时间(或闪烁时间), 单位为秒。

- 表示行人闪烁黄灯提示期间;
- 期间禁止新行人进入, 允许已进入人行道者安全通过;
- 通常设置为行人平均步速约 1.0-1.2 m/s 与横断面宽度的比值。"

::= { phaseCfgEntry 37 }

4.3.2.4.3 相位控制表

phaseCtlTable OBJECT-TYPE

SYNTAX	SEQUENCE OF phaseCtlEntry
ACCESS	not-accessible
STATUS	optional
DESCRIPTION	

"包含每个相位控制的信息数据组, 数组元素数量等于maxPhase。"

::= { phase 3 }

phaseCtlEntry OBJECT-TYPE

SYNTAX	phaseCtlEntry
ACCESS	not-accessible
STATUS	optional
DESCRIPTION	

"每个相位的控制数据组。这是一个数据序列。"

INDEX	{ index }
-------	-----------

::= { phaseCtlTable 1 }

phaseCtlEntry ::= SEQUENCE{
 index INTEGER,

```
mask          INTEGER,
Disable       INTEGER}
```

4.3.2.4.3.1 相位编号

```
index OBJECT-TYPE
    SYNTAX      INTEGER(1..64)
    ACCESS       read-write
    STATUS       optional
    DESCRIPTION
        "信号机内关于相位的序号，整数型，相位1的编号为1，以此类推，最大64个
        相位。"
 ::= { phaseCtlEntry 1 }
```

4.3.2.4.3.2 相位屏蔽

```
mask OBJECT-TYPE
    SYNTAX      INTEGER(
                        0 (Mask_OFF),
                        1 (Mask_ON))
    ACCESS       read-write
    STATUS       optional
    DESCRIPTION
        "信号机内相位屏蔽控制。
        0：表示该相位正常输出；
        1：表示屏蔽该相位输出（该相位所有输出通道强制变成红灯）"
 ::= { phaseCtlEntry 2 }
```

4.3.2.4.3.3 相位禁止

```
disable OBJECT-TYPE
    SYNTAX      INTEGER(
                        0 (Disable_OFF),
                        1 (Disable_ON))
    ACCESS       read-write
    STATUS       optional
    DESCRIPTION
        "信号机内相位禁止控制。
        0：表示该相位正常输出；
        1：表示禁止该相位输出（该相位所有输出通道强制变成灭灯）。
        相位禁止的优先级高于相位屏蔽。"
 ::= { phaseCtlEntry 3 }
```

4.3.2.4.4 相位状态表

```
phaseStateTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF phaseStateEntry
    ACCESS       not-accessible
    STATUS       mandatory
    DESCRIPTION
        "包含每个相位状态的信息数据组，数组元素数量等于maxPhase。"
 ::= { phase 4 }
```

```
phaseStateEntry OBJECT-TYPE
    SYNTAX      phaseStateEntry
    ACCESS       not-accessible
    STATUS       mandatory
    DESCRIPTION
        "每个相位的状态数据组。这是一个数据序列。"
    INDEX       { index }
 ::= { phaseStateTable 1 }
```

```
phaseStateEntry ::= SEQUENCE{
    index          INTEGER,
```

```

state          INTEGER,
elapsedTime    INTEGER,
remainderTime  INTEGER}

```

4.3.2.4.4.1 相位编号

index OBJECT-TYPE

```

SYNTAX          INTEGER(1..64)
ACCESS          read-only
STATUS          mandatory
DESCRIPTION

```

“信号机内关于相位的序号，整数型，相位1的编号为1，以此类推，最大64个相位。”

```
 ::= { phaseStateEntry 1 }
```

4.3.2.4.4.2 相位状态

state OBJECT-TYPE

```

SYNTAX          INTEGER(
                    0x10(Status_NotOfWay),
                    0x20(Status_OnTheWay),
                    0x30(Status_Transition))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION

```

“相位的状态，定义如下：

a) 未放行 (Status_NotOfWay)	0x10
b) 正在放行 (Status_OnTheWay)	0x20
c) 过渡 (Status_Transition)	0x30

```
 ::= { phaseStateEntry 2 }
```

4.3.2.4.4.3 相位已经运行的时间

elapsedTime OBJECT-TYPE

```

SYNTAX          INTEGER(0..65535)
ACCESS          read-only
STATUS          mandatory
DESCRIPTION

```

“相位已运行的时间，整数型，单位：1s。”

```
 ::= { phaseStateEntry 3 }
```

4.3.2.4.4.4 相位剩余时间感应等不确定结束时间点的控制模式，读出的值如何理解

remainderTime OBJECT-TYPE

```

SYNTAX          INTEGER(0..65535)
ACCESS          read-only
STATUS          optional
DESCRIPTION

```

“相位剩余的时间，整数型，单位：1s。”

```
 ::= { phaseStateEntry 4 }
```

4.3.2.5 数据类5 检测器信息

```
detector OBJECT IDENTIFIER ::= { asc 5 }
```

-- 定义检测器信息节点

-- 此节点中包含与信号控制机检测器信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.检测器信息(1.3.6.1.4.1.61332.3.2.2.5)。

4.3.2.5.1 实际检测器数

maxDetector OBJECT-TYPE

```

SYNTAX          INTEGER(0..128)
ACCESS          read-only
STATUS          mandatory

```

DESCRIPTION

“该信号机实际使用的检测器编号的最大值，整数型，范围0~128。”

```
::= { detector 1 }
```

4.3.2.5.2 检测器配置表

detectorCfgTable OBJECT-TYPE

SYNTAX SEQUENCE OF detectorCfgEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“包含每个检测器配置的信息数据组，数组元素数量等于maxDetector。”

```
::= { detector 2 }
```

detectorCfgEntry OBJECT-TYPE

SYNTAX detectorCfgEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“每个检测器的信息数据组。这是一个数据序列。”

INDEX { index }

```
::= { detectorCfgTable 1 }
```

detectorCfgEntry ::= SEQUENCE{

index INTEGER,

type INTEGER,

samplePeriod1 INTEGER,

samplePeriod2 INTEGER,

info OCTET STRING,

intersectionIndex INTEGER,

laneID INTEGER,

Position INTEGER,

targetType INTEGER,

dataType INTEGER}

4.3.2.5.2.1 检测器编号

index OBJECT-TYPE

SYNTAX INTEGER(1..128)

ACCESS read-only

STATUS mandatory

DESCRIPTION

“信号机内关于检测器的序号，整数型，检测器1的编号为1，以此类推，最大128个检测器。”

```
::= { detectorCfgEntry 1 }
```

4.3.2.5.2.2 检测器类型

type OBJECT-TYPE

SYNTAX

INTEGER(

0(Detector_Type_Unavailable),

1(Detector_Type_Loop),

2(Detector_Type_Video),

3(Detector_Type_Geomagnetic),

4(Detector_Type_Microwave),

5(Detector_Type_Ultrasonic),

6(Detector_Type_Infrared),

7(Detector_Type_RFID),

8(Detector_Type_Laser),

9(Detector_Type_MultiTargetRadar),

10(Detector_Type_RadarVideo),

11(Detector_Type_VideoRFID),

12(Detector_Type_PedestrianButton),

255(Detector_Type_Undefined))

ACCESS read-write
STATUS mandatory
DESCRIPTION

“检测器的类型，定义如下：

a) 未配置 (Detector_Type_Unavailable)	0x00
b) 线圈 (Detector_Type_Loop)	0x01
c) 视频 (Detector_Type_Video)	0x02
d) 地磁 (Detector_Type_Geomagnetic)	0x03
e) 微波断面检测器 (Detector_Type_Microwave)	0x04
f) 超声波检测器 (Detector_Type_Ultrasonic)	0x05
g) 红外检测器 (Detector_Type_Infrared)	0x06
h) RFID 检测器 (Detector_Type_RFID)	0x07
i) 激光检测器 (Detector_Type_Laser)	0x08
j) 多目标雷达检测 (Detector_Type_MultiTargetRadar)	0x09
k) 雷达视频一体机 (Detector_Type_RadarVideo)	0x0A
l) 视频RFID一体机 (Detector_Type_VideoRFID)	0x0B
m) 行人按钮 (Detector_Type_PedestrianButton)	0x0C
n) 本标准未定义的类型 (Detector_Type_Undefined)	0xFF”

::= { detectorCfgEntry 2 }

4.3.2.5.2.3 检测器采集周期1/流量采集周期

samplePeriod1 OBJECT-TYPE
SYNTAX INTEGER(0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION

“检测器采集周期1，单位：1s。

当设置为0时，应当解释为当数据变化时立即发送，当设置为65535时，应当解释为根据信号控制周期发送。

为实现前向兼容，也定义为检测器流量采集周期，单位：1s。

当设置为0时，应当解释为当数据变化时立即发送，当设置为65535时，应当解释为根据信号控制周期发送。”

::= { detectorCfgEntry 3 }

4.3.2.5.2.4 检测器采集周期2/占有率采集周期

samplePeriod2 OBJECT-TYPE
SYNTAX INTEGER(0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION

“检测器采集周期2，单位：1s。

当设置为0时，应当解释为当数据变化时立即发送，当设置为65535时，应当解释为根据信号控制周期发送。

为实现前向兼容，也定义为检测器占有率采集周期，单位：1s。

当设置为0时，应当解释为当数据变化时立即发送，当设置为65535时，应当解释为根据信号控制周期发送。”

::= { detectorCfgEntry 4 }

4.3.2.5.2.5 检测器安装位置描述

info OBJECT-TYPE
SYNTAX OCTET STRING(SIZE(0..128))
ACCESS read-write
STATUS optional
DESCRIPTION

“检测器的安装位置或者其他必要信息的具体位置的文字说明，字符不满128字节以0填充。”

::= { detectorCfgEntry 5 }

4.3.2.5.2.6 检测器归属路口序号

intersectionIndex OBJECT-TYPE
 SYNTAX INTEGER(1..8)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “检测器归属的路口序号。”
 ::= { detectorCfgEntry 6 }

4.3.2.5.2.7 检测器归属车道号

laneID OBJECT-TYPE
 SYNTAX INTEGER(0..128)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “检测器对应的车道编号，同一车道的检测器归属车道编号应当一致。
 该数据目前用于当一个车道存在多个检测器时，通过该属性信息确定相互关联关系。
 例如：如果车道编号1的车道配置了三个检测器，分别是停车线前截面检测器，远端排队溢出检测器，以及停车线前区域检测器，这几个检测器的归属车道号需要保持一致。”
 ::= { detectorCfgEntry 7 }

4.3.2.5.2.8 检测器检测位置

position OBJECT-TYPE
 SYNTAX INTEGER(
 0x00 (Detector_Pos_Unavailable),
 0x01 (Detector_Pos_BeforeStopLine),
 0x02 (Detector_Pos_AfterStopLine),
 0x03 (Detector_Pos_UpstreamQueneOverflow),
 0x04 (Detector_Pos_DownstreamQueneOverflow),
 0x05 (Detector_Pos_Preempt),
 0x11 (Detector_Pos_Area_UpstreamLane),
 0x12 (Detector_Pos_Area_DownstreamLane),
 0x21 (Detector_Pos_PedestrianButton),
 0x22 (Detector_Pos_Area_PedestrianWait),
 0x23 (Detector_Pos_PedestrianCrossing),
 0x24 (Detector_Pos_Area_PedestrianCrossing),
 0xFF (Detector_Pos_Undefined))
 ACCESS read-write
 STATUS optional
 DESCRIPTION

“检测器的检测位置，定义如下：

- | | |
|---|------|
| a) 未配置 (Detector_Pos_Unavailable) | 0x00 |
| b) 停止线前 (Detector_Pos_BeforeStopLine) | 0x01 |
| 该位置对于机动车/非机动车车道，对应停车线前位置，一般要求等候车辆在等候期间能够被持续检测到。 | |
| c) 停止线后 (Detector_Pos_AfterStopLine) | 0x02 |
| 该位置对于机动车/非机动车车道，对应通过停车线后位置，一般要求等候车辆不能触发该检测器，越线停车车辆可以被检测到。 | |
| d) 上游排队溢出检测 (Detector_Pos_UpstreamQueneOverflow) | 0x03 |
| 该位置对于机动车/非机动车车道，对应车道上游一定距离的位置，当车辆持续占有该位置，认为排队长度已经触发某逻辑门限。 | |
| e) 下游排队溢出检测 (Detector_Pos_DownstreamQueneOverflow) | 0x04 |
| 该位置对于机动车/非机动车车道，对应下游入口，当持续检测车辆存在，认为下游已经发生车辆排队溢出。 | |
| f) 远端优先检测 (Detector_Pos_Preempt) | 0x05 |

该检测器通常设置在机动车道上游一个合理距离，用于检测公交车辆或者其他有优先请求的车辆，该检测器被触发，应当调用对应的优先控制逻辑。

- g) 车道停止线前区域 (Detector_Pos_Area_UpstreamLane) 0x11
该区域通常覆盖停止线前至车辆来向远端一定距离的范围，通常采用雷达、视频类设备，提供车辆来向排队、个体位置、速度等信息。
- h) 下游车道入口区域 (Detector_Pos_Area_DownstreamLane) 0x12
该区域通常覆盖下游入口至车辆去向远端一定距离的范围，通常采用雷达、视频类设备，提供下游入口车辆排队、个体位置、速度等信息。
- i) 行人按钮 (Detector_Pos_PedestrianButton) 0x21
该检测器对应行人过街请求按钮信号。
- j) 行人等候区 (Detector_Pos_Area_PedestrianWait) 0x22
该检测区域对应路缘石以上的行人等候区，一般用于检测是否有人等待、人员数量以及等候时间。
- k) 人行横道截面 (Detector_Pos_PedestrianCrossing) 0x23
该位置对应人行横道一个断面，用于统计整体的上下行人流量，设置位置一般为便于统计上下游人流数量的合理位置。
- l) 人行横道区 (Detector_Pos_Area_PedestrianCrossing) 0x24
该检测区域对应人行横道区域，一般用于检测是否有人在人行横道通行，人员的空间占有率等。
- m) 未定义 (Detector_Pos_Undefined) 0xFF
本协议目前未定义的检测范围。”
::={ detectorCfgEntry 8 }

4.3.2.5.2.9 检测器检测对象

targetType OBJECT-TYPE

SYNTAX INTEGER(0..65535)

ACCESS read-write

STATUS optional

DESCRIPTION

“检测器的检测对象，定义为一个16位无符号数，其中每个Bit对应一种检测对象类型，检测器可以同时支持多个检测对象类型，此时只要将对应Bit位置为1，不支持的对应Bit位置为0，各Bit位检测对象定义如下：

Bit 0 - 机动车 (Detect_Target_Vehicle 0x01)

Bit 1 - 非机动车 (Detect_Target_NonVehicle 0x02)

Bit 2 - 行人 (Detect_Target_Pedestrian 0x04)

Bit 3 - 公交车 (Detect_Target_Bus 0x08)

Bit 4 - 有轨电车 (Detect_Target_Tram 0x10)

Bit 5 - 紧急车辆 (Detect_Target_Emergency 0x20)

Bit 6 ~ Bit 15 - 保留”

::= { detectorCfgEntry 9 }

4.3.2.5.2.10 检测器检测数据类型

dataType OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS optional

DESCRIPTION

“检测器的检测数据类型，定义为一个32位无符号数，其中每个Bit对应一种检测数据类型，检测器可以同时支持多个检测数据类型，此时只要将对应Bit位置为1，不支持的对应Bit位置为0，各Bit位检测数据对象定义如下：

Bit 0 - 存在状态/行人请求状态 (Detector_Data_State)

Bit 1 - 速度 (Detector_Data_Speed)

Bit 2 - 类型 (Detector_Data_Type)

Bit 3 - 身份 (Detector_Data_ID)

Bit 4 - 排队长度 (Detector_Data_QueneLength)

Bit 5 - 排队数量 (Detecor_Data_QueneNum)

Bit 6 - 等候数量 (Detector_Data_WaitNum)
 Bit 7 - 存在数量 (Detector_Data_TargetNum)
 Bit 8 - 目标状态列表 (Detector_Data_TargetList)
 Bit 9 ~ Bit 15 - 保留
 Bit 16 - 流量 (Detector_Stat_Flow)
 Bit 17 - 平均占有率 (Detector_Stat_Occupancy)
 Bit 18 - 平均车速 (Detector_Stat_Speed)
 Bit 19 - 平均车头时距 (Detector_Stat_TimeHeadWay)
 Bit 20 - 平均车间时距 (Detector_Stat_TimeGap)
 Bit 21 - 平均停车次数 (Detector_Stat_StopTimes)
 Bit 22 - 平均排队长度 (Detector_Stat_QueneLength)
 Bit 23 - 最大排队长度 (Detector_Stat_MaxQueneLength)
 Bit 24 - 平均等候人数 (Detector_Stat_PedWaitNum)
 Bit 25 - 平均等候时间 (Detector_Stat_PedWaitTime)
 Bit 26 - 最大等候人数 (Detector_Stat_MaxPedWaitNum)
 Bit 27 - 最大等候时间 (Detector_Stat_MaxPedWaitTime)
 Bit 28 ~ Bit 31 - 保留”
 ::= { detectorCfgEntry 10 }

4.3.2.5.3 检测器数据状态表

detectorDataTable OBJECT-TYPE
 SYNTAX SEQUENCE OF detectorDataEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 “包含每个检测器数据的信息数据组，数组元素数量等于maxDetector。”
 ::= { detector 3 }

detectorDataEntry OBJECT-TYPE
 SYNTAX detectorDataEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 “每个检测器数据的信息数据组。这是一个数据序列。”
 INDEX { index }
 ::= { detectorDataTable 1 }

detectorDataEntry ::= SEQUENCE{
 index INTEGER,
 sampleTime Counter64,
 state INTEGER,
 targetSpeed INTEGER,
 targetType INTEGER,
 targetID OCTET STRING,
 queneLength INTEGER,
 targetNum INTEGER,
 targetTotalNum INTEGER,
 targetDataTable SEQUENCE OF targetDataEntry}

4.3.2.5.3.1 检测器编号

index OBJECT-TYPE
 SYNTAX INTEGER(1..128)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “信号机内关于检测器的序号，整数型，检测器1的编号为1，以此类推，最大128个检测器。”
 ::= { detectorDataEntry 1 }

4.3.2.5.3.2 检测器状态数据生成时间

sampleTime OBJECT-TYPE
 SYNTAX Counter64 (0..9999999999999)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “相对1970年1月1日0时0分0毫秒的毫秒计时，13位的十进制数。
 当采用RAW格式时，时间为8字节（Int64），高字节在前，低字节在后。
 当采用JSON格式时，时间为13位的整形数据。”
 ::= { detectorDataEntry 2 }

4.3.2.5.3.3 存在状态/行人请求状态

state OBJECT-TYPE
 SYNTAX INTEGER(0,1)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “0：没有对象存在/没有行人请求，1：有对象存在/有行人请求。”
 ::= { detectorDataEntry 3 }

4.3.2.5.3.4 检测对象的速度

targetSpeed OBJECT-TYPE
 SYNTAX INTEGER(0..255)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 “当前通过检测器的检测对象的速度，单位：公里/小时”
 ::= { detectorDataEntry 4 }

4.3.2.5.3.5 检测对象的类型

targetType OBJECT-TYPE
 SYNTAX INTEGER(
 0(Target_Unavailable),
 1(Target_Vehicle_Small),
 2(Target_Vehicle_Middle),
 3(Target_Vehicle_Large),
 4(Target_Bus),
 5(Target_Tram),
 6(Target_Special),
 16(Target_Bicycle),
 17(Target_Electric_Bicycle),
 255(Target_Undefined))
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 “最新的通过检测器的对象类型，检测器检出目标定义如下：
 a) 无目标 (Target_Unavailable) 0x00
 b) 小型车 (车长<6 m) (Target_Vehicle_Small) 0x01
 c) 中型车 (6 m≤车长<12 m) (Target_Vehicle_Middle) 0x02
 d) 大型车 (车长≥12 m) (Target_Vehicle_Large) 0x03
 e) 公交车 (Target_Bus) 0x04
 f) 有轨电车 (Target_Tram) 0x05
 g) 特种车辆 (Target_Special) 0x06
 h) 自行车 (Target_Bicycle) 0x10
 i) 电动自行车 (Target_Electric_Bicycle) 0x11
 j) 其他 (Target_Undefined) 0xFF”
 ::= { detectorDataEntry 5 }

4.3.2.5.3.6 检测对象的身份

targetID OBJECT-TYPE

SYNTAX OCTET STRING(SIZE(1..16))
 ACCESS read-only
 STATUS optional
 DESCRIPTION

“最新的通过检测器的对象的身份，对于机动车辆，为号牌识别信息，对于其他类型的目标，应当是具备唯一性的目标ID。”

::= { detectorDataEntry 6 }

4.3.2.5.3.7 排队长度

queueLength OBJECT-TYPE

SYNTAX INTEGER(0..65535)
 ACCESS read-only
 STATUS optional
 DESCRIPTION

“检测器的检测区域的车辆排队长度（单位：米）。”

::= { detectorDataEntry 7 }

4.3.2.5.3.8 排队数量/等候数量/行人数量

targetNum OBJECT-TYPE

SYNTAX INTEGER(0..65535)
 ACCESS read-only
 STATUS optional
 DESCRIPTION

“对于机动车检测器，对应检测器检测到的车道排队车辆数量；
 对于行人等候区检测器，对应行人等候区的行人等候数量；
 对于人行横道检测器，对应人行横道区域的行人数量。”

::= { detectorDataEntry 8 }

4.3.2.5.3.9 检测区域目标总数

targetTotalNum OBJECT-TYPE

SYNTAX INTEGER(0..65535)
 ACCESS read-only
 STATUS optional
 DESCRIPTION

“检测器检测到的所有目标总数。”

::= { detectorDataEntry 9 }

4.3.2.5.3.10 检测区域目标列表

targetDataTable OBJECT-TYPE

SYNTAX SEQUENCE OF targetDataEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION

“包含检测器目标详细数据的信息数据组，数组元素数量等于
 targetTotalNum。”

::= { detectorDataEntry 10 }

targetDataEntry OBJECT-TYPE

SYNTAX targetDataEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION

“每个检测器目标数据的信息数据组。这是一个数据序列。”

INDEX {index}

::= { targetDataTable 1 }

targetDataEntry ::= SEQUENCE{

index INTEGER,
 type INTEGER,
 speed INTEGER,
 position INTEGER}

4.3.2.5.3.10.1 目标编号

index OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-only

STATUS optional

DESCRIPTION

“检测器目标的序号，整数型，目标1的编号为1，以此类推，最大255个目标。”

::= { targetDataEntry 1 }

4.3.2.5.3.10.2 目标类型

type OBJECT-TYPE

SYNTAX

INTEGER(

0(Target_Unavailable),

1(Target_Vehicle_Small),

2(Target_Vehicle_Middle),

3(Target_Vehicle_Large),

4(Target_Bus),

5(Target_Tram),

6(Target_Special),

16(Target_Bicycle),

17(Target_Electric_Bicycle),

255(Target_Undefined))

ACCESS read-only

STATUS optional

DESCRIPTION

“检测器目标的类型，整数型，参见对象4.3.2.5.3.5检测对象的类型定义。”

::= { targetDataEntry 2 }

4.3.2.5.3.10.3 目标速度

speed OBJECT-TYPE

SYNTAX INTEGER(-256..255)

ACCESS read-only

STATUS optional

DESCRIPTION

“检测器目标的速度，整数型，单位：千米/小时，远离停车线为正，接近停车线为负。”

::={ targetDataEntry 3 }

4.3.2.5.3.10.4 目标位置

position OBJECT-TYPE

SYNTAX INTEGER(-32768..32767)

ACCESS read-only

STATUS optional

DESCRIPTION

“车辆位置相对停车线的距离，单位：0.1米。远离路口方向为正，路口范围内为负。”

::= { targetDataEntry 4 }

4.3.2.5.4 检测器控制表

detectorCtlTable OBJECT-TYPE

SYNTAX SEQUENCE OF detectorCtlEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

“包含每个检测器控制的信息数据组，数组元素数量等于maxDetector。”

::= { detector 4 }

detectorCtlEntry OBJECT-TYPE

SYNTAX detectorCtlEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

“每个检测器控制的信息数据组。这是一个数据序列。”

INDEX { index }

::= { detectorCtlEntry 1 }

detectorCtlEntry ::= SEQUENCE{

index INTEGER,

enable INTEGER,

restart INTEGER,

trapEnable INTEGER,

trapPeriod INTEGER}

4.3.2.5.4.1 检测器编号

index OBJECT-TYPE

SYNTAX INTEGER(1..128)

ACCESS read-only

STATUS optional

DESCRIPTION

“信号机内关于检测器的序号，整数型，检测器1的编号为1，以此类推，最大128个检测器。”

::= { detectorCtlEntry 1 }

4.3.2.5.4.2 检测器开关与4.3.2.5.4.4 检测器开关 中文名称重叠

enable OBJECT-TYPE

SYNTAX INTEGER(0,1)

ACCESS read-write

STATUS optional

DESCRIPTION

“检测器使能控制开关。

值为0：表示该检测器关闭，关闭后，检测器应当停止发送交通数据。

值为1：表示该检测器启用。”

::= { detectorCtlEntry 2 }

4.3.2.5.4.3 检测器重启

restart OBJECT-TYPE

SYNTAX INTEGER(0,1)

ACCESS read-write

STATUS optional

DESCRIPTION

“检测器重启控制开关。

写入1：触发检测器重启；

写入0：无动作；

读取1：检测器重启过程中；

读取0：检测器重启完成。”

::= { detectorCtlEntry 3 }

4.3.2.5.4.4 检测器开关

trapEnable OBJECT-TYPE

SYNTAX INTEGER(0,1)

ACCESS read-write

STATUS optional

DESCRIPTION

“检测器数据主动上传使能控制开关。

值为0：表示该检测器数据主动上传关闭。

值为1：表示该检测器数据主动上传启用。”

::= { detectorCtlEntry 4 }

4.3.2.5.4.5 检测器数据主动上传周期

trapPeriod OBJECT-TYPE

SYNTAX INTEGER(0..65535)

```

ACCESS          read-write
STATUS          optional
DESCRIPTION
    “检测器数据主动上传周期，单位秒。
    设为0代表检测到数据变化立即上传；
    设为65535应当解释为根据信号周期主动上传。”
 ::= { detectorCtlEntry 5 }

```

4.3.2.6 数据类6 阶段信息

```

stage OBJECT IDENTIFIER ::= { asc 6 }
-- 定义阶段信息节点
-- 此节点中包含与信号控制机阶段信息相关的所有数据对象
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.阶段信息(1.3.6.1.4.1.61332.3.2.2.6)。

```

4.3.2.6.1 实际配置阶段数

```

maxStage OBJECT-TYPE
    SYNTAX          INTEGER(0..64)
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        “信号机中已经使用的最大的阶段编号。
        当信号机配置使用环逻辑时，此项参数将被忽略，此时应当将数据配置为0。”
 ::= { stage 1 }

```

4.3.2.6.2 阶段配置表

```

stageCfgTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF stageCfgEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION
        “包含每个阶段配置的信息数据组，数组元素数量等于maxStage。”
 ::= { stage 2 }

```

```

stageCfgEntry OBJECT-TYPE
    SYNTAX          stageCfgEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION
        “每个阶段配置的信息数据组。这是一个数据序列。”
    INDEX          { index }
 ::= { stageCfgTable 1 }

```

```

stageCfgEntry ::= SEQUENCE{
    index          INTEGER,
    includedPhase  SEQUENCE OF INTEGER}

```

4.3.2.6.2.1 阶段编号

```

index OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        “信号机中阶段的序号，整数型，阶段1的序号为1，以此类推，最大64个阶段。
        ”
 ::= { stageCfgEntry 1 }

```

4.3.2.6.2.2 阶段的相位

```

includedPhase OBJECT-TYPE

```

SYNTAX SEQUENCE OF INTEGER(SIZE(1..64))
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION

“当采用RAW格式时，阶段的相位长度固定为8个字节(64 bit)，bit0代表相位1，bit63代表相位64，数据结构为 BYTE8、...BYTE1，代表 bit63～bit56、...bit7～bit0，相应的 bit 位为1时，表示该相位包含在该阶段中。当采用JSON格式时，阶段的相位为包含1~64个数字的数组，每个数字代表该阶段包含的一个相位的编号。”

::= { stageCfgEntry 2 }

4.3.2.6.3 阶段状态表

stageStateTable OBJECT-TYPE

SYNTAX SEQUENCE OF stageStateEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION

“包含每个阶段状态的信息数据组，数组元素数量等于maxStage。”

::= { stage 3 }

stageStateEntry OBJECT-TYPE

SYNTAX stageStateEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION

“每个阶段状态的信息数据组。这是一个数据序列。”

INDEX {index}

::= { stageStateTable 1 }

stageStateEntry ::= SEQUENCE{

index INTEGER,
 state INTEGER,
 elapsedTime INTEGER,
 remainderTime INTEGER}

4.3.2.6.3.1 阶段编号

index OBJECT-TYPE

SYNTAX INTEGER(1..64)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION

“信号机中阶段的序号，整数型，阶段1的序号为1，以此类推，最大64个阶段。”

::= { stageStateEntry 1 }

4.3.2.6.3.2 阶段状态

state OBJECT-TYPE

SYNTAX INTEGER(
 0x10(Status_NotOfWay),
 0x20(Status_OnTheWay),
 0x30(Status_Transition))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION

“阶段的状态，定义如下：

- | | |
|---------------------------|------|
| a) 未放行 (Status_NotOfWay) | 0x10 |
| b) 正在放行 (Status_OnTheWay) | 0x20 |
| c) 过渡 (Status_Transition) | 0x30 |

::= { stageStateEntry 2 }

4.3.2.6.3.3 阶段已经运行的时间

elapsedTime OBJECT-TYPE
 SYNTAX INTEGER(0..65535)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “阶段已运行的时间，整数型，单位：1s。”
 ::= { stageStateEntry 3 }

4.3.2.6.3.4 阶段剩余时间感应等不确定结束时间点的控制模式，读出的值如何理解

remainderTime OBJECT-TYPE
 SYNTAX INTEGER(0..65535)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 “阶段剩余的时间，整数型，单位：1s。”
 ::= { stageStateEntry 4 }

4.3.2.6.4 阶段控制表

stageCtlTable OBJECT-TYPE
 SYNTAX SEQUENCE OF stageCtlEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 “包含每个阶段控制的信息数据组，数组元素数量等于maxStage。”
 ::= { stage 4 }

stageCtlEntry OBJECT-TYPE
 SYNTAX stageCtlEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 “每个阶段控制的信息数据组。这是一个数据序列。”
 INDEX { index }
 ::= { stageCtlTable 1 }

stageCtlEntry ::= SEQUENCE{
 index INTEGER,
 softwareRequest INTEGER,
 mask INTEGER,
 disable INTEGER}

4.3.2.6.4.1 阶段编号

index OBJECT-TYPE
 SYNTAX INTEGER(1..64)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 “信号机中阶段的序号，整数型，阶段1的序号为1，以此类推，最大64个阶段。”
 “
 ::= { stageCtlEntry 1 }

4.3.2.6.4.2 阶段软件需求

softwareRequest OBJECT-TYPE
 SYNTAX INTEGER(0,1)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “阶段的软件需求，一般用于实现强制保持一个阶段的需求状态。
 0：该阶段没有软件需求；
 1：该阶段有软件需求，软件需求即中心通过通信报文设置阶段的检测器请求。”

```
::= { stageCtlEntry 2 }
```

4.3.2.6.4.3 阶段屏蔽标志

```
mask OBJECT-TYPE
```

```
SYNTAX          INTEGER(0,1)
```

```
ACCESS          read-write
```

```
STATUS          optional
```

```
DESCRIPTION
```

“该阶段是否屏蔽。

0：不屏蔽；

1：屏蔽，即中心通过通信报文屏蔽某个阶段，一旦某个阶段被屏蔽了则阶段链中将不会出现该阶段（该阶段的所有相位为红灯）。”

```
::= { stageCtlEntry 3 }
```

4.3.2.6.4.4 阶段禁止标志

```
disable OBJECT-TYPE
```

```
SYNTAX          INTEGER(0,1)
```

```
ACCESS          read-write
```

```
STATUS          optional
```

```
DESCRIPTION
```

“该阶段是否禁止输出。

0：不禁止；

1：禁止，一旦某个阶段被禁止了，则阶段链中将不会出现该阶段（该阶段的所有相位为灭灯）”

```
::= { stageCtlEntry 4 }
```

4.3.2.7 数据类7 相位安全信息

```
safe OBJECT IDENTIFIER ::= { asc 7 }
```

-- 定义相位安全信息节点

-- 此节点中包含与信号控制机相位安全信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.相位安全信息(1.3.6.1.4.1.61332.3.2.2.7)。

4.3.2.7.1 相位冲突配置表

```
conflictTable OBJECT-TYPE
```

```
SYNTAX          SEQUENCE OF conflictEntry
```

```
ACCESS          not-accessible
```

```
STATUS          mandatory
```

```
DESCRIPTION
```

“包含每个相位冲突配置的信息数据组，数组元素数量等于64。”

```
::= { safe 1 }
```

```
conflictEntry OBJECT-TYPE
```

```
SYNTAX          conflictEntry
```

```
ACCESS          not-accessible
```

```
STATUS          mandatory
```

```
DESCRIPTION
```

“每个相位冲突配置的信息数据组。这是一个数据序列。”

```
INDEX          { index }
```

```
::= { conflictTable 1 }
```

```
conflictEntry ::= SEQUENCE{
```

```
index          INTEGER,
```

```
conflictList   SEQUENCE OF INTEGER}
```

4.3.2.7.1.1 相位编号

```
index OBJECT-TYPE
```

```
SYNTAX          INTEGER(1..64)
```

```
ACCESS          read-only
```

STATUS mandatory

DESCRIPTION

“信号机中相位的序号，整数型，相位1的序号为1，以此类推，最大64个相位。”

::= { conflictEntry 1 }

4.3.2.7.1.2 冲突相位序列

conflictList OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..64))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“当采用RAW格式时，冲突相位序列的长度为8个字节(64 bit)，bit0代表相位1，bit63代表相位64，数据结构为BYTE8~BYTE1，代表 bit63~bit56、...bit7~bit0，相应的bit位为1时，表示该相位和该属性中相应编号的相位冲突。

当采用JSON格式时，冲突相位序列为一个包含0到64个数字的数组，数组中每个数字代表一个和该相位冲突的相位的编号。”

::= { conflictEntry 2 }

4.3.2.7.2 相位绿间隔配置表

greenIntervalTable OBJECT-TYPE

SYNTAX SEQUENCE OF greenIntervalEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“包含每个相位过渡绿间隔配置的信息数据组，数组元素数量等于64。”

::= { safe 2 }

greenIntervalEntry OBJECT-TYPE

SYNTAX greenIntervalEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“每个相位过渡绿间隔配置的信息数据组。这是一个数据序列。”

INDEX { index }

::= { greenIntervalTable 1 }

greenIntervalEntry ::= SEQUENCE{

index INTEGER,

intervalList SEQUENCE OF INTEGER}

4.3.2.7.2.1 相位编号

index OBJECT-TYPE

SYNTAX INTEGER(1..64)

ACCESS read-only

STATUS mandatory

DESCRIPTION

“信号机中相位的序号，整数型，相位1的序号为1，以此类推，最大64个相位。”

::={ greenIntervalEntry 1 }

4.3.2.7.2.2 绿间隔时间序列

intervalList OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..64))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“当采用RAW格式时，绿间隔时间序列是一个64字节的数组，该数组中，Byte63~Byte0分别代表从当前相位过渡到目标相位所需要经过的绿间隔时间，

Byte0对应目标相位1, Byte1对应目标相位2, ..., Byte63对应目标相位64, 单位: 0.1 s。

当采用JSON格式时, 绿间隔时间序列是一个包含64个整型数字的数组, 该数组中每个数字代表从当前相位过渡到该数字所对应的目标相位所需要经过的绿间隔时间, INTEGER[0]对应目标相位1, INTEGER[1]对应目标相位2, ..., INTEGER[63]对应目标相位64, 单位: 0.1 s。”

```
::= { greenIntervalEntry 2 }
```

4.3.2.7.3 相位并发配置表

concurrentTable OBJECT-TYPE

SYNTAX SEQUENCE OF concurrentEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

“包含每个相位并发配置的信息数据组, 数组元素数量等于64。

本表适用于基于相位的控制模型(如环控型信号机), 支持通过并发配置定义各相位之间的兼容通行关系。

对于采用阶段控制结构的信号机, 该表项可为空或忽略使用, 建议使用阶段-相位映射结构(如 stagePhaseTable)代替。”

```
::= { safe 3 }
```

concurrentEntry OBJECT-TYPE

SYNTAX concurrentEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

“每个相位并发配置的信息数据组。这是一个数据序列。”

INDEX { index }

```
::= { concurrentTable 1 }
```

concurrentEntry ::= SEQUENCE{

index INTEGER,

concurrentList SEQUENCE OF INTEGER}

4.3.2.7.3.1 相位编号

index OBJECT-TYPE

SYNTAX INTEGER(1..64)

ACCESS read-only

STATUS optional

DESCRIPTION

“信号机中相位的序号, 整数型, 相位1的序号为1, 以此类推, 最大64个相位。”

”

```
::= { concurrentEntry 1 }
```

4.3.2.7.3.2 并发相位序列

concurrentList OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..64))

ACCESS read-write

STATUS optional

DESCRIPTION

“当采用RAW格式时, 相位的并发相位序列长度固定为8个字节(64 bit), bit0代表相位1, bit63代表相位64, 数据结构为 BYTE8、...BYTE1, 代表 bit63~bit56、...bit7~bit0, 相应的 bit 位为1时, 表示该相位包含在该相位的并发相位中。

当采用JSON格式时, 相位的并发相位序列为一个包含0到64个整型数字的数组, 该数组中每个数字代表一个对应的并发相位的编号。”

```
::= { concurrentEntry 2 }
```

4.3.2.7.4 相位关联配置表

```

conjunctionTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF conjunctionEntry
    ACCESS          not-accessible
    STATUS          optional
    DESCRIPTION
        "包含每个相位关联配置的信息数据组，数组元素数量等于64。"
 ::= { safe 4 }

```

```

conjunctionEntry OBJECT-TYPE
    SYNTAX          conjunctionEntry
    ACCESS          not-accessible
    STATUS          optional
    DESCRIPTION
        "每个相位关联配置的信息数据组。这是一个数据序列。"
    INDEX          { index }
 ::= { conjunctionTable 1 }

```

```

conjunctionEntry ::= SEQUENCE{
    index          INTEGER,
    conjunctionList SEQUENCE OF INTEGER}

```

4.3.2.7.4.1 相位编号

```

index OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS          read-only
    STATUS          optional
    DESCRIPTION
        "信号机中相位的序号，整数型，相位1的序号为1，以此类推，最大64个相位。"
 ::= { conjunctionEntry 1 }

```

4.3.2.7.4.2 关联相位序列

```

conjunctionList OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(0..64))
    ACCESS          read-write
    STATUS          optional
    DESCRIPTION
        "相位的关联相位是指多个相位整体上采取同步操作且相互之间存在制约关系的。
        例如同方向的机动车相位和行人相位可以构成相位关联，此时机动车相位的中止需要同步考虑行人相位的参数限制（例如行人清空时间）。NTCIP的相位参数同时包含了机动车和行人的参数，而本协议的机动车相位和行人相位是分开定义，难以实现相互的关联，因此定义本数据属性。

        当采用RAW格式时，相位的关联相位序列长度固定为8个字节(64 bit)，bit0代表相位1，bit63代表相位64，数据结构为 BYTE8、...BYTE1，代表 bit63~bit56、...bit7~bit0，相应的 bit 位为1时，表示该相位包含在该相位的关联相位中。

        当采用JSON格式时，相位的关联相位序列为一个包含0到64个Number的数组，该数组中每个Number代表一个对应的关联相位的编号。

        注意：同一个相位不可以出现在多个相位关联组中。"
 ::= { conjunctionEntry 2 }

```

4.3.2.8 数据类8 紧急优先信息

```

preempt OBJECT IDENTIFIER ::= { asc 8 }
-- 定义紧急优先信息节点
-- 此节点中包含与信号控制机紧急优先信息相关的所有数据对象

```

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
 iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
 设备.交通信号控制机.紧急优先信息(1.3.6.1.4.1.61332.3.2.2.8)。

4.3.2.8.1 实际优先数量

```
maxPreempt OBJECT-TYPE
    SYNTAX          INTEGER(0..64)
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "该信号机实际使用的优先编号的最大值，整数型，范围0~64。"
 ::= { preempt 1 }
```

4.3.2.8.2 优先配置表

```
preemptCfgTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF preemptCfgEntry
    ACCESS           not-accessible
    STATUS           optional
    DESCRIPTION
        "包含每个优先配置的信息数据组，数组元素数量等于maxPreempt。"
 ::= { preempt 2 }
```

```
preemptCfgEntry OBJECT-TYPE
    SYNTAX          preemptCfgEntry
    ACCESS           not-accessible
    STATUS           optional
    DESCRIPTION
        "每个优先的信息数据组。这是一个数据序列。"
    INDEX           { index }
 ::= { preemptCfgTable 1 }
```

```
preemptCfgEntry ::= SEQUENCE{
    index          INTEGER,
    callStage      INTEGER,
    callPhase      SEQUENCE OF INTEGER,
    priority       INTEGER,
    mask           INTEGER,
    source         INTEGER,
    rspInterval    INTEGER}
```

4.3.2.8.2.1 优先编号

```
index OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "信号机内关于优先的序号，整数型，优先1的编号为1，以此类推，最大64个  

        优先。"
 ::= { preemptCfgEntry 1 }
```

4.3.2.8.2.2 优先申请阶段

```
callStage OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "当采用阶段语义时，对应优先申请的阶段序号；  

        当采用环语义时，本标识号无意义。"
 ::= { preemptCfgEntry 2 }
```

4.3.2.8.2.3 优先申请相位

callPhase OBJECT-TYPE
 SYNTAX SEQUENCE OF INTEGER(SIZE(1..64))
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “当采用环语义时，是一个包含1到64个整型数字的数组，每个数组元素对应一个优先申请的相位序号；
 当采用阶段语义时，本标识号无意义。”
 ::= { preemptCfgEntry 3 }

4.3.2.8.2.4 优先申请优先级

piority OBJECT-TYPE
 SYNTAX INTEGER(0..255)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “优先信号的优先级，数值越小优先级越高，0的优先级最高。”
 ::= { preemptCfgEntry 3 }

4.3.2.8.2.5 优先屏蔽标志

mask OBJECT-TYPE
 SYNTAX INTEGER(0,1)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “0：不屏蔽；1：屏蔽。被屏蔽的优先将不被响应。”
 ::= { preemptCfgEntry 4 }

4.3.2.8.2.6 优先来源

source OBJECT-TYPE
 SYNTAX INTEGER(1..128)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “优先对应检测器的编号。”
 ::= { preemptCfgEntry 6 }

4.3.2.8.2.7 优先响应间隔

rspInterval OBJECT-TYPE
 SYNTAX INTEGER(0..65535)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “优先响应间隔时间设置，单位：秒。
 当连续两次优先请求时间间隔小于此参数设定时，后一次优先请求将会暂时不处理。如果时间间隔达到，且优先请求仍然保持时，该优先将被响应。”
 ::= { preemptCfgEntry 7 }

4.3.2.8.3 优先状态表

preemptStateTable OBJECT-TYPE
 SYNTAX SEQUENCE OF preemptStateEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 “包含每个优先状态的信息数据组，数组元素数量等于maxPreempt。”
 ::= { preempt 3 }

preemptStateEntry OBJECT-TYPE
 SYNTAX preemptStateEntry
 ACCESS not-accessible
 STATUS optional

DESCRIPTION

“每个优先状态的信息数据组。这是一个数据序列。”

INDEX { index }

::= { preemptStateTable 1 }

preemptStateEntry ::= SEQUENCE{

index INTEGER,

requestState INTEGER,

executeState INTEGER,

intervalCnt Counter}

4.3.2.8.3.1 优先编号

index OBJECT-TYPE

SYNTAX INTEGER(1..64)

ACCESS read-only

STATUS optional

DESCRIPTION

“信号机内关于优先的序号，整数型，优先1的编号为1，以此类推，最大64个优先。”

::= { preemptStateEntry 1 }

4.3.2.8.3.2 优先申请状态

requestState OBJECT-TYPE

SYNTAX INTEGER(0,1)

ACCESS read-only

STATUS optional

DESCRIPTION

“0：没有申请，1：有申请。”

::= { preemptStateEntry 2 }

4.3.2.8.3.3 优先执行状态

executeState OBJECT-TYPE

SYNTAX INTEGER(
0(idle),
1(run))

ACCESS read-only

STATUS optional

DESCRIPTION

“0：没有执行，1：执行。”

::= { preemptStateEntry 3 }

4.3.2.8.3.4 优先间隔计时

intervalCnt OBJECT-TYPE

SYNTAX Counter(0..65535)

ACCESS read-only

STATUS optional

DESCRIPTION

“当一个优先请求被系统响应，则该计时数值设置为优先响应间隔对应数值，然后按时间每秒进行减一操作，直到数据归零。该数据值为零，代表可以响应下一次优先请求。”

::= { preemptStateEntry 4 }

4.3.2.8.4 实际紧急数量

maxUrgent OBJECT-TYPE

SYNTAX INTEGER(0..64)

ACCESS read-write

STATUS optional

DESCRIPTION

“该信号机实际使用的紧急编号的最大值，整数型，范围0~64。”

::= { preempt 4 }

4.3.2.8.5 紧急配置表

```

urgentCfgTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF urgentCfgEntry
    ACCESS           not-accessible
    STATUS           optional
    DESCRIPTION
        "包含每个紧急配置的信息数据组，数组元素数量等于maxUrgent。"
 ::= { preempt 5 }

urgentCfgEntry OBJECT-TYPE
    SYNTAX          urgentCfgEntry
    ACCESS           not-accessible
    STATUS           optional
    DESCRIPTION
        "每个紧急的信息数据组。这是一个数据序列。"
    INDEX           { index }
 ::= { urgentCfgTable 1 }

urgentCfgEntry ::= SEQUENCE{
    index           INTEGER,
    callStage       INTEGER,
    callPhase       SEQUENCE OF INTEGER,
    priority        INTEGER,
    mask            INTEGER,
    source          INTEGER,
    rspInterval     INTEGER}

```

4.3.2.8.5.1 紧急编号

```

index OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "信号机内关于紧急的序号，整数型，紧急1的编号为1，以此类推，最大64个紧急。"
 ::= { urgentCfgEntry 1 }

```

4.3.2.8.5.2 紧急申请阶段

```

callStage OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "当采用阶段语义时，对应紧急申请的阶段序号；
        当采用环语义时，该标识号无意义。"
 ::= { urgentCfgEntry 2 }

```

4.3.2.8.5.3 紧急申请相位

```

callPhase OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(1..64))
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "当采用环语义时，是一个包含1到64个整型数字的数组，每个数组元素对应一个紧急申请的相位序号；
        当采用阶段语义时，本标识号无意义。"
 ::= { urgentCfgEntry 3 }

```

4.3.2.8.5.4 紧急申请优先级

```

priority OBJECT-TYPE
    SYNTAX          INTEGER(0..255)

```

ACCESS read-write
 STATUS optional
 DESCRIPTION

“紧急信号的优先级，数值越小优先级越高，0的优先级最高。”

::= { urgentCfgEntry 4 }

4.3.2.8.5.5 紧急屏蔽标志

mask OBJECT-TYPE

SYNTAX INTEGER(0,1)
 ACCESS read-write
 STATUS optional
 DESCRIPTION

“0：不屏蔽；1：屏蔽。”

::= { urgentCfgEntry 4 }

4.3.2.8.5.6 紧急来源

source OBJECT-TYPE

SYNTAX INTEGER(1..128)
 ACCESS read-write
 STATUS optional
 DESCRIPTION

“紧急对应检测器的编号。”

::= { urgentCfgEntry 6 }

4.3.2.8.5.7 紧急响应间隔

rspInterval OBJECT-TYPE

SYNTAX INTEGER(0..65535)
 ACCESS read-write
 STATUS optional
 DESCRIPTION

“紧急响应间隔时间设置，单位：秒。

当连续两次紧急请求时间间隔小于此参数设定时，后一次紧急请求将会暂时不处理。如果时间间隔达到，且紧急请求仍然保持时，该紧急将被响应。”

::= { urgentCfgEntry 7 }

4.3.2.8.6 紧急状态表

urgentStateTable OBJECT-TYPE

SYNTAX SEQUENCE OF urgentStateEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION

“包含每个紧急状态的信息数据组，数组元素数量等于maxUrgent。”

::= { preempt 6 }

urgentStateEntry OBJECT-TYPE

SYNTAX urgentStateEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION

“每个紧急状态的信息数据组。这是一个数据序列。”

INDEX { index }

::= { urgentStateTable 1 }

urgentStateEntry ::= SEQUENCE{

index INTEGER,
 requestState INTEGER,
 executeState INTEGER,
 intervalCnt Counter}

4.3.2.8.6.1 紧急编号

index OBJECT-TYPE

SYNTAX INTEGER(1..64)
ACCESS read-only
STATUS optional
DESCRIPTION

“信号机内关于紧急的序号，整数型，紧急1的编号为1，以此类推，最大64个紧急。”

::= { urgentStateEntry 1 }

4.3.2.8.6.2 紧急申请状态

requestState OBJECT-TYPE

SYNTAX INTEGER(0,1)
ACCESS read-only
STATUS optional
DESCRIPTION

“0：没有申请，1：有申请。”

::= { urgentStateEntry 2 }

4.3.2.8.6.3 紧急执行状态

executeState OBJECT-TYPE

SYNTAX INTEGER(
0 (Idle),
1 (Run))
ACCESS read-only
STATUS optional
DESCRIPTION

“0：没有执行，1：执行。”

::= { urgentStateEntry 3 }

4.3.2.8.6.4 紧急时间间隔

intervalCnt OBJECT-TYPE

SYNTAX Counter(0..65535)
ACCESS read-only
STATUS optional
DESCRIPTION

“当一个紧急请求被系统响应，则该计时数值设置为紧急响应间隔对应数值，然后按时间每秒进行减一操作，直到数据归零。该数据值为零，代表可以响应下一次紧急请求。”

::= { urgentStateEntry 4 }

4.3.2.9 数据类9 阶段方案信息

stagePattern OBJECT IDENTIFIER ::= { asc 9 }

-- 定义阶段方案信息节点

-- 此节点中包含与信号控制机阶段方案信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.阶段方案信息(1.3.6.1.4.1.61332.3.2.2.9)。

4.3.2.9.1 实际方案数量

maxPattern OBJECT-TYPE

SYNTAX INTEGER(0..128)
ACCESS read-write
STATUS mandatory
DESCRIPTION

“该信号机实际使用的方案编号的最大值，整数型，范围0~128。

当使用环交通语义时，此项设置为0。”

::= { stagePattern 1 }

4.3.2.9.2 方案配置表

patternCfgTable OBJECT-TYPE

SYNTAX SEQUENCE OF patternCfgEntry

```

ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION

```

“包含每个方案配置的信息数据组，数组元素数量等于maxPattern。”

```

::= { stagePattern 2 }

```

```

patternCfgEntry OBJECT-TYPE

```

```

SYNTAX          patternCfgEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION

```

“每个方案的信息数据组。这是一个数据序列。”

```

INDEX          { index }

```

```

::= { patternCfgTable 1 }

```

```

patternCfgEntry ::= SEQUENCE{

```

```

    index          INTEGER,
    cycle          INTEGER,
    coordinationIndex  INTEGER,
    offset          INTEGER,
    offsetSyncPoint  INTEGER,
    stageChain      SEQUENCE OF INTEGER,
    stageTimeChain  SEQUENCE OF INTEGER,
    stageTypeChain  SEQUENCE OF INTEGER,
    lateBeginTimeChain SEQUENCE OF (SEQUENCE OF INTEGER),
    earlyEndTimeChain SEQUENCE OF (SEQUENCE OF INTEGER),
    minGreenChain   SEQUENCE OF INTEGER,
    maxGreen1Chain  SEQUENCE OF INTEGER,
    maxGreen2Chain  SEQUENCE OF INTEGER,
    extendedGreenChain SEQUENCE OF INTEGER,
    detectorChain   SEQUENCE OF (SEQUENCE OF INTEGER),
    maxWasteChain   SEQUENCE OF INTEGER,
    lossOnSwitchChain SEQUENCE OF INTEGER)

```

4.3.2.9.2.1 方案编号

```

index OBJECT-TYPE

```

```

SYNTAX          INTEGER(1..128)
ACCESS          read-only
STATUS          mandatory
DESCRIPTION

```

“信号机内关于阶段方案的序号，整数型，方案1的编号为1，以此类推，最大128个方案。”

```

::= { patternCfgEntry 1 }

```

4.3.2.9.2.2 方案周期

```

cycle OBJECT-TYPE

```

```

SYNTAX          INTEGER(1..65535)
ACCESS          read-write
STATUS          mandatory
DESCRIPTION

```

“阶段方案运行周期，单位：1s。”

```

::= { patternCfgEntry 2 }

```

4.3.2.9.2.3 方案协调序号

```

coordinationIndex OBJECT-TYPE

```

```

SYNTAX          INTEGER(1..16)
ACCESS          read-write
STATUS          optional
DESCRIPTION

```

“阶段方案的阶段链中需要协调的阶段序号。”

```

::= { patternCfgEntry 3 }

```

4.3.2.9.2.4 方案相位差时间

```
offset OBJECT-TYPE
    SYNTAX          INTEGER(0..65535)
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "相位差时间，单位：1s。"
 ::= { patternCfgEntry 4 }
```

4.3.2.9.2.5 方案相位差计算参考点阶段方案中并未定义”协调相位“，建议改为第一个出现的相位或是最后一个出现的相位等

```
offsetSyncPoint OBJECT-TYPE
    SYNTAX          INTEGER(
        0(Offset_Ref_Unavailable),
        1(Offset_Ref_GreenOn),
        2(Offset_Ref_GreenOff),
        3(Offset_Ref_YellowOff))
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "相位差计算参考点类型定义如下：
        a) 未设置 (Offset_Ref_Unavailable)                0x00
        b) 相对协调相位绿灯开始 (Offset_Ref_GreenOn)      0x01
        c) 相对协调相位绿灯结束 (Offset_Ref_GreenOff)     0x02
        d) 相对协调相位黄灯结束 (Offset_Ref_YellowOff)    0x03
 ::= { patternCfgEntry 5 }
```

4.3.2.9.2.6 方案阶段链

```
stageChain OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(1..16))
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "通信协议规定：阶段方案的阶段链中阶段的数量最大为16。
        当采用RAW格式时，阶段方案的阶段链为一个16字节的数组，结构为：BYTE0、
        BYTE1、...BYTE15，BYTE0的值代表方案中第一个放行的阶段号，以此类推
        BYTE15的值代表方案中放行的第16个阶段，不满16个阶段用0填充相应的位（
        即0表示不存在的阶段）。
        当采用JSON格式时，阶段方案的阶段链为一个包含1到16个整型数字的数组，
        数组中每个数字代表一个阶段编号，数组中的顺序代表方案中阶段的执行顺序。
        "
 ::= { patternCfgEntry 6 }
```

4.3.2.9.2.7 方案阶段时间链

```
stageTimeChain OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(1..16))
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "当采用RAW格式时，阶段方案的阶段时间链为一个32字节的数组，每个阶段的
        时间占用2个字节，结构为：BYTE0、BYTE1、...BYTE31，BYTE0~BYTE1（
        多字节表示一个数值时，高字节在前，低字节在后）的值代表阶段1的放行时间，
        以此类推直到阶段16，单位：1s。
        当采用JSON格式时，阶段方案的阶段时间链为一个包含1到16个整型数字的数
        组，数组中每个数字代表对应阶段的放行时间，注意阶段方案的阶段时间链与
        阶段方案的阶段链数组长度应当一致。"
 ::= { patternCfgEntry 7 }
```

4.3.2.9.2.8 方案阶段出现类型链

stageTypeChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“Array 数组中的每个元素代表阶段方案的阶段链相同数组索引下阶段出现的类型，阶段出现类型定义如下：

a) 固定出现 (Type_Fix) 0x10

b) 按需求出现 (Type_Demand) 0x20

注：阶段出现类型规定阶段链中阶段出现的类型：固定出现是指在每个信号周期内都会出现；按需求出现是指在每个信号周期内根据其检测器上的需求确定是否出现。

当采用RAW格式时，阶段方案的阶段出现类型链为一个16字节的数组，BYTE0代表阶段方案的阶段链中BYTE0对应阶段出现的类型，以此类推BYTE15代表阶段方案的阶段链中BYTE15对应的阶段出现类型。

当采用JSON格式时，阶段方案的阶段出现类型链为一个包含1到16个整型数字的数组，数组中每个数字代表阶段方案的阶段链中对应阶段的出现类型，注意阶段方案的阶段出现类型链与阶段方案的阶段链应当数组长度一致。”

::= { patternCfgEntry 8 }

4.3.2.9.2.9 方案阶段相位晚启动时间链

lateBeginTimeChain OBJECT-TYPE

SYNTAX SEQUENCE OF (SEQUENCE OF INTEGER(SIZE(1..64)))
(SIZE(1..16))

ACCESS read-write

STATUS optional

DESCRIPTION

“2维数组，代表16个阶段，每个阶段针对64个相位单独指定晚启动时间，单位：1s。

当采用RAW格式时，每个阶段的相位晚启动时间需求占用64个字节，Byte0代表相位1晚启动时间，Byte63代表相位64晚启动时间，数据结构为BYTE63、...、BYTE0，则方案的阶段链相位晚启动时间的长度为1024个字节，结构为：

BYTE0、BYTE1、...BYTE1023， BYTE0~BYTE63对应阶段1的相位晚启动时间，以此类推直到阶段16。

当采用JSON格式时，阶段方案的阶段相位晚启动时间链为一个2维数组，分别对应阶段方案的阶段以及阶段的相位晚启动时间，注意应当与阶段方案的阶段数量、阶段中相位数量保持一致。”

::= { patternCfgEntry 9 }

4.3.2.9.2.10 方案阶段相位早结束时间链

earlyEndTimeChain OBJECT-TYPE

SYNTAX SEQUENCE OF (SEQUENCE OF INTEGER(SIZE(1..64)))
(SIZE(1..16))

ACCESS read-write

STATUS optional

DESCRIPTION

“2维数组，代表16个阶段，每个阶段针对64个相位单独指定早结束时间，单位：1s。

当采用RAW格式时，每个阶段的相位早结束时间需求占用64个字节，Byte0代表相位1早结束时间，Byte63代表相位64早结束时间，数据结构为BYTE63、...、BYTE0，则方案的阶段链相位早结束时间的长度为1024个字节，结构为：

BYTE0、BYTE1、...BYTE1023， BYTE0~BYTE63对应阶段1的相位早结束时间，以此类推直到阶段16。

当采用JSON格式时，阶段方案的阶段相位早结束时间链为一个2维数组，分别对应阶段方案的阶段以及该阶段的相位早结束时间，注意应当与阶段方案的阶段数量、阶段中相位数量保持一致。”

```
::= { patternCfgEntry 10 }
```

4.3.2.9.2.11 方案阶段最小绿灯时间链

minGreenChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“当采用RAW格式时，阶段方案的阶段最小绿灯时间链为一个32字节的数组，每个阶段的最小绿灯时间占用2个字节，结构为：BYTE0、BYTE1、...BYTE31，BYTE0～BYTE1（多字节表示一个数值时，高字节在前，低字节在后）的值代表阶段1的最小绿灯时间，以此类推直到阶段16，单位：1s。

当采用JSON格式时，阶段方案的阶段最小绿灯时间链为一个包含1到16个整型数字的数组，数组中每个数字代表对应阶段的阶段最小绿灯时间，注意阶段方案的阶段最小绿灯时间链与阶段方案的阶段链数组长度应当一致。”

```
::= { patternCfgEntry 11 }
```

4.3.2.9.2.12 方案阶段最大绿灯时间1链

maxGreen1Chain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“当采用RAW格式时，阶段方案的阶段最大绿灯时间1链为一个32字节的数组，每个阶段的阶段最大绿灯时间1占用2个字节，结构为：BYTE0、BYTE1、...BYTE31，BYTE0～BYTE1（多字节表示一个数值时，高字节在前，低字节在后）的值代表阶段1的最大绿灯时间1，以此类推直到阶段16，单位：1s。

当采用JSON格式时，阶段方案的阶段最大绿灯时间1链为一个包含1到16个整型数字的数组，数组中每个数字代表对应阶段的阶段最大绿灯时间1，注意阶段方案的阶段最大绿灯时间1链与阶段方案的阶段链数组长度应当一致。”

```
::= { patternCfgEntry 12 }
```

4.3.2.9.2.13 方案阶段最大绿灯时间2链

maxGreen2Chain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“当采用RAW格式时，阶段方案的阶段最大绿灯时间2链为一个32字节的数组，每个阶段的阶段最大绿灯时间2占用2个字节，结构为：BYTE0、BYTE1、...BYTE31，BYTE0～BYTE1（多字节表示一个数值时，高字节在前，低字节在后）的值代表阶段1的最大绿灯时间2，以此类推直到阶段16，单位：1s。

当采用JSON格式时，阶段方案的阶段最大绿灯时间2链为一个包含1到16个整型数字的数组，数组中每个数字代表对应阶段的阶段最大绿灯时间2，注意阶段方案的阶段最大绿灯时间2链与阶段方案的阶段链数组长度应当一致。”

```
::= { patternCfgEntry 13 }
```

4.3.2.9.2.14 方案阶段单位延长绿时间链

extendedGreenChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“当采用RAW格式时，阶段方案的阶段单位延长绿时间链为一个32字节的数组，每个阶段的单位延长绿时间占用2个字节，结构为：BYTE0、BYTE1、...BYTE31，BYTE0～BYTE1（多字节表示一个数值时，高字节在前，低字节在后）的值代表阶段1的单位延长绿时间，以此类推直到阶段16，单位：1s。

当采用JSON格式时，阶段方案的阶段单位延长绿灯时间链为一个包含1到16个整型数字的数组，数组中每个数字代表对应阶段的单位延长绿灯时间，注意阶段方案的阶段单位延长绿灯时间链与阶段方案的阶段链数组长度应当一致。”

```
::= { patternCfgEntry 14 }
```

4.3.2.9.2.15 方案阶段相位感应链

detectorChain OBJECT-TYPE

SYNTAX SEQUENCE OF (SEQUENCE OF INTEGER(SIZE(1..64))
(SIZE(1..16))

ACCESS read-write

STATUS optional

DESCRIPTION

“当采用RAW格式时，阶段方案的阶段相位感应链是一个包含16个64Bit数据的数组，每个阶段的相位感应占用8个字节（64 bit），bit0代表相位1，bit63代表相位64，数据结构为BYTE7、...、BYTE0，代表bit63~bit56、...、bit7~bit0，相应的bit位为1时，表示该相位对应检测器上存在车辆会请求该阶段，则方案的阶段链需求的长度为128个字节，结构为：BYTE0、BYTE1、...BYTE127，BYTE0~BYTE7对应阶段方案中阶段1的需求，以此类推直到阶段16，不满16个阶段用 0 填充相应的位置。

当采用JSON格式时，阶段方案的阶段相位感应链是一个2维数组，分别对应阶段方案的阶段以及阶段包含的相位中参与感应的相位编号，注意如果参与感应的相位编号并不包含在阶段的相位中，只要该相位配置了检测器，则检测器的数据依然应当对阶段感应逻辑产生作用。”

```
::= { patternCfgEntry 15 }
```

4.3.2.9.2.16 方案阶段最大浪费时间链

maxWasteChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS optional

DESCRIPTION

“Array 数组中的每两个字节代表感应控制模式下本方案阶段最大浪费时间，单位：1s。

注意JSON格式数据每个时间对应一个数据，不区分高低字节。

当采用RAW格式时，阶段方案的阶段最大浪费时间链为一个32字节的数组，每个阶段的最大浪费时间占用2个字节，结构为：BYTE0、BYTE1、...BYTE31，BYTE0~BYTE1（多字节表示一个数值时，高字节在前，低字节在后）的值代表阶段1的最大浪费时间，以此类推直到阶段16，单位：1s。

当采用JSON格式时，阶段方案的阶段最大浪费时间链为一个包含1到16个整型数字的数组，数组中每个数字代表对应阶段的最大浪费时间，注意阶段方案的阶段最大浪费时间链与阶段方案的阶段链数组长度应当一致。”

```
::= { patternCfgEntry 16 }
```

4.3.2.9.2.17 方案阶段切换损失时间链

lossOnSwitchChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS optional

DESCRIPTION

“该数据对应绿灯初期和黄灯末期的损失时间之和。对于一个相位被切分在多个阶段的，需要根据阶段中相位实际处在的运行状态确定损失时间，例如该阶段对应相位启动，则损失时间包含绿灯初期的损失时间，如果该阶段包含相位结束时间，则该损失时间应当包含黄灯末期的损失，如果不包含上述两种情况中任何一种，则损失时间为0。

当采用RAW格式时，阶段方案的阶段切换损失时间链为一个32字节的数组，每个阶段的切换损失时间占用2个字节，结构为：BYTE0、BYTE1、...BYTE31，

BYTE0~BYTE1（多字节表示一个数值时，高字节在前，低字节在后）的值代表阶段1的切换损失时间，以此类推直到阶段16，单位：1s。

当采用JSON格式时，阶段方案的阶段切换损失时间链为一个包含1到16个整型数字的数组，数组中每个数字代表对应阶段的切换损失时间，注意阶段方案的阶段切换损失时间链与阶段方案的阶段链数组长度应当一致。”

```
::= { patternCfgEntry 17 }
```

4.3.2.10 数据类10 过渡约束信息

```
constrain OBJECT IDENTIFIER ::= { asc 10 }
-- 定义过渡约束信息节点
-- 此节点中包含与信号控制机过渡约束信息相关的所有数据对象
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.过渡约束信息(1.3.6.1.4.1.61332.3.2.2.10)。
```

4.3.2.10.1 阶段过渡约束配置表

```
stageContrainTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF stageContrainEntry
    ACCESS           not-accessible
    STATUS           mandatory
    DESCRIPTION
        "包含每个阶段过渡约束配置的信息数据组，数组元素数量等于64。"
    ::= { constrain 1 }
```

```
stageContrainEntry OBJECT-TYPE
    SYNTAX          stageContrainEntry
    ACCESS           not-accessible
    STATUS           mandatory
    DESCRIPTION
        "每个阶段过渡约束配置的信息数据组。这是一个数据序列。"
    INDEX           { index }
    ::= { stageContrainTable 1 }
```

```
stageContrainEntry ::= SEQUENCE{
    index          INTEGER,
    constrain      SEQUENCE OF INTEGER}
```

4.3.2.10.1.1 目标阶段编号

```
index OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "信号机中阶段的序号，阶段1的阶段编号固定为1，以此类推。当信号机采用
        阶段语义时，该数据属性生效。"
    ::= { stageContrainEntry 1 }
```

4.3.2.10.1.2 过渡约束值

```
contrain OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(64))
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "当采用阶段语义时，代表阶段过渡的约束值；当采用环语义时，无意义。
        当采用RAW格式时，阶段过渡约束值为长度为64个字节的数组，结构为BYTE63
        ~BYTE0，BYTE63对应阶段64，BYTE0对应阶段1。
        当采用JSON格式时，阶段过渡约束值为含64个整型数字的数组，每个数字对应
        相应的阶段。
        值为0：表示可以直接跳转；
```

值不为0且不等于255：表示不能直接跳转，该值表示需要经过该阶段号；
 值为255：表示不能跳转。”

```
::= { stageContrainEntry 2 }
```

4.3.2.10.2 相位过渡约束配置表

```
phaseContrainTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF phaseContrainEntry
    ACCESS           not-accessible
    STATUS           optional
    DESCRIPTION
        "包含每个相位过渡约束配置的信息数据组，数组元素数量等于64。"
::= { constrain 2 }
```

```
phaseContrainEntry OBJECT-TYPE
    SYNTAX          phaseContrainEntry
    ACCESS           not-accessible
    STATUS           optional
    DESCRIPTION
        "每个相位过渡约束配置的信息数据组。这是一个数据序列。"
    INDEX           {index}
::= { phaseContrainTable 1 }
```

```
phaseContrainEntry ::= SEQUENCE{
    index          INTEGER,
    constrain      SEQUENCE OF INTEGER}
```

4.3.2.10.2.1 目标相位编号

```
index OBJECT-TYPE
    SYNTAX          INTEGER(1..64)
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "信号机中相位的序号，相位1的相位编号固定为1，以此类推。当信号机采用  

        环语义时，该数据属性生效。"
::= { phaseContrainEntry 1 }
```

4.3.2.10.2.2 过渡约束值

```
constrain OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(64))
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "当采用环语义时，代表相位过渡的约束值；当采用阶段语义时，无意义。  

        当采用RAW格式时，相位过渡约束值为长度为64个字节的数组，结构为BYTE63  

        ~BYTE0，BYTE63对应相位64，BYTE0对应相位1。  

        当采用JSON格式时，相位过渡约束值为含64个Number的数组，每个Number  

        对应相应的相位。  

        值为0：表示可以直接跳转；  

        值不为0且不等于255：表示不能直接跳转，该值表示需要经过该相位号；  

        值为255：表示不能跳转。"
::= { phaseContrainEntry 2 }
```

4.3.2.11 数据类11 日计划信息

```
dayPlan OBJECT IDENTIFIER ::= { asc 11 }
-- 定义日计划信息节点
-- 此节点中包含与信号控制机日计划信息相关的所有数据对象
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.日计划信息(1.3.6.1.4.1.61332.3.2.2.11)。
```

4.3.2.11.1 实际日计划数量

maxDayPlan OBJECT-TYPE
 SYNTAX INTEGER(0..128)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “该信号机实际使用的日计划编号的最大值，整数型，范围0~128。”
 ::= { dayPlan 1 }

4.3.2.11.2 日计划配置表

dayPlanCfgTable OBJECT-TYPE
 SYNTAX SEQUENCE OF dayPlanCfgEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 “包含每个日计划配置的信息数据组，数组元素数量等于maxDayPlan。”
 ::= { dayPlan 2 }

dayPlanCfgEntry OBJECT-TYPE
 SYNTAX dayPlanCfgEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 “每个日计划的信息数据组。这是一个数据序列。”
 INDEX { index }
 ::= { dayPlanCfgTable 1 }

dayPlanCfgEntry ::= SEQUENCE{
 index INTEGER,
 beginTimeChain SEQUENCE OF INTEGER,
 stagePatternChain SEQUENCE OF INTEGER,
 ringPatternChain SEQUENCE OF INTEGER,
 runModeChain SEQUENCE OF INTEGER,
 actionChain1 SEQUENCE OF INTEGER,
 actionChain2 SEQUENCE OF INTEGER,
 actionChain3 SEQUENCE OF INTEGER,
 actionChain4 SEQUENCE OF INTEGER,
 actionChain5 SEQUENCE OF INTEGER,
 actionChain6 SEQUENCE OF INTEGER,
 actionChain7 SEQUENCE OF INTEGER,
 actionChain8 SEQUENCE OF INTEGER}

4.3.2.11.2.1 日计划编号

index OBJECT-TYPE
 SYNTAX INTEGER(1..128)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “信号机中日计划的序号，日计划1的编号为1，以此类推。”
 ::= { dayPlanCfgEntry 1 }

4.3.2.11.2.2 时段开始时间链

beginTimeChain OBJECT-TYPE
 SYNTAX SEQUENCE OF INTEGER(SIZE(0..96))
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 “日计划详细时段设置，协议约定一天最多48个时段，
 Byte Array[1]:Byte Array[0]表示每天的开始时段，
 Byte Array[1]表示时钟，Byte Array[0]表示分钟，

时段1固定为0:00（即每天从0:00开始），以此类推，直到第48个时段。

当采用RAW格式时，为固定96个字节的数组；

当采用JSON格式时，为包含多个字节的数组，每两个字节对应一个时段，时段总数可变。”

```
::= { dayPlanCfgEntry 2 }
```

4.3.2.11.2.3 时段执行阶段方案链

stagePatternChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..48))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“Byte Array[0]的值表示当天的第1个时段执行的方案，以此类推直到第48个时段。注意当采用阶段语义时，方案对应阶段方案，当采用环语义时，本标识号无意义。”

```
::= { dayPlanCfgEntry 3 }
```

4.3.2.11.2.4 时段执行环方案链

ringPatternChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..48))

ACCESS read-write

STATUS optional

DESCRIPTION

“Byte Array[0]的值表示当天的第1个时段执行的方案，以此类推直到第48个时段。注意当采用环语义时，方案对应环方案，当采用阶段语义时，本标识号无意义。”

```
::= { dayPlanCfgEntry 4 }
```

4.3.2.11.2.5 时段运行模式链

runModeChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..48))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“Byte Array[0]的值表示当天的第1个时段信号机运行的模式，以此类推直到第48个时段。模式定义详见附录A.4”

```
::= { dayPlanCfgEntry 5 }
```

4.3.2.11.2.6 时段动作链1

actionChain1 OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..96))

ACCESS read-write

STATUS optional

DESCRIPTION

“每天的时段动作链1，用于描述信号机在相应时段的动作，
Byte Array[1]:Byte Array[0]表示每天的第一个时段的动作，
Byte Array[1]表示动作类型，Byte Array[0]表示动作参数，以此类推。”

```
::= { dayPlanCfgEntry 6 }
```

4.3.2.11.2.7 时段动作链2

actionChain2 OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..96))

ACCESS read-write

STATUS optional

DESCRIPTION

“每天的时段动作链2，用于描述信号机在相应时段的动作，
Byte Array[1]:Byte Array[0]表示每天的第一个时段的动作，
Byte Array[1]表示动作类型，Byte Array[0]表示动作参数，以此类推。”

```
::= { dayPlanCfgEntry 7 }
```

4.3.2.11.2.8 时段动作链3

```

actionChain3 OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(0..96))
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "每天的时段动作链3，用于描述信号机在相应时段的动作，
        Byte Array[1]:Byte Array[0]表示每天的第一个时段的动作，
        Byte Array[1]表示动作类型，Byte Array[0]表示动作参数，以此类推。"
    ::= { dayPlanCfgEntry 8 }

```

4.3.2.11.2.9 时段动作链4

```

actionChain4 OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(0..96))
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "每天的时段动作链4，用于描述信号机在相应时段的动作，
        Byte Array[1]:Byte Array[0]表示每天的第一个时段的动作，
        Byte Array[1]表示动作类型，Byte Array[0]表示动作参数，以此类推。"
    ::= { dayPlanCfgEntry 9 }

```

4.3.2.11.2.10 时段动作链5

```

actionChain5 OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(0..96))
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "每天的时段动作链5，用于描述信号机在相应时段的动作，
        Byte Array[1]:Byte Array[0]表示每天的第一个时段的动作，
        Byte Array[1]表示动作类型，Byte Array[0]表示动作参数，以此类推。"
    ::= { dayPlanCfgEntry 10 }

```

4.3.2.11.2.11 时段动作链6

```

actionChain6 OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(0..96))
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "每天的时段动作链6，用于描述信号机在相应时段的动作，
        Byte Array[1]:Byte Array[0]表示每天的第一个时段的动作，
        Byte Array[1]表示动作类型，Byte Array[0]表示动作参数，以此类推。"
    ::= { dayPlanCfgEntry 11 }

```

4.3.2.11.2.12 时段动作链7

```

actionChain7 OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(0..96))
    ACCESS           read-write
    STATUS           optional
    DESCRIPTION
        "每天的时段动作链7，用于描述信号机在相应时段的动作，
        Byte Array[1]:Byte Array[0]表示每天的第一个时段的动作，
        Byte Array[1]表示动作类型，Byte Array[0]表示动作参数，以此类推。"
    ::= { dayPlanCfgEntry 12 }

```

4.3.2.11.2.13 时段动作链8

```

actionChain8 OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(0..96))
    ACCESS           read-write
    STATUS           optional

```

DESCRIPTION

“每天的时段动作链8，用于描述信号机在相应时段的动作，
 Byte Array[1]:Byte Array[0]表示每天的第一个时段的动作，
 Byte Array[1]表示动作类型，Byte Array[0]表示动作参数，以此类推。”
 ::= { dayPlanCfgEntry 13 }

4.3.2.12 数据类12 调度信息

schedule OBJECT IDENTIFIER ::= { asc 12 }
 -- 定义调度信息节点
 -- 此节点中包含与信号控制机调度信息相关的所有数据对象
 -- 当使用SNMP协议访问本数据类时，设备标识号路径为：
 iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
 设备.交通信号控制机.调度信息(1.3.6.1.4.1.61332.3.2.2.12)。

4.3.2.12.1 实际调度表数量

maxSchedule OBJECT-TYPE
 SYNTAX INTEGER(0..128)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “该信号机实际使用的调度表编号的最大值，整数型，范围0~128。”
 ::= { schedule 1 }

4.3.2.12.2 调度配置表

scheduleCfgTable OBJECT-TYPE
 SYNTAX SEQUENCE OF scheduleCfgEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 “包含每个调度配置的信息数据组，数组元素数量等于maxSchedule。”
 ::= { schedule 2 }

scheduleCfgEntry OBJECT-TYPE
 SYNTAX scheduleCfgEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 “每个调度配置的信息数据组。这是一个数据序列。”
 INDEX { index }
 ::= { scheduleCfgTable 1 }

scheduleCfgEntry ::= SEQUENCE{
 index INTEGER,
 priority INTEGER,
 weekday INTEGER,
 month INTEGER,
 date INTEGER,
 dayPlan INTEGER}

4.3.2.12.2.1 调度表编号

index OBJECT-TYPE
 SYNTAX INTEGER(1..128)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “信号机中调度表的序号，调度表1的编号为1，以此类推。”
 ::= { scheduleCfgEntry 1 }

4.3.2.12.2.2 调度表优先级

priority OBJECT-TYPE

```

SYNTAX          INTEGER(0..255)
ACCESS          read-write
STATUS          mandatory
DESCRIPTION

```

“调度表的优先级，值越小优先级越高，0的优先级最高。”

```
 ::= { scheduleCfgEntry 2 }
```

4.3.2.12.2.3 调度表星期值

```

day OBJECT-TYPE
    SYNTAX          INTEGER(0..255)
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION

```

“bit0表示星期天，当bit0为1时，表示该调度表星期天需要执行，bit1为星期一，以此类推，bit7 保留。”

```
 ::= { scheduleCfgEntry 3 }
```

4.3.2.12.2.4 调度表月份值

```

month OBJECT-TYPE
    SYNTAX          INTEGER(0..65535)
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION

```

“16位整型数。Bit0保留，当bit1为1时，表示该调度表1月需要执行，以此类推，bit13~bit15保留。”

```
 ::= { scheduleCfgEntry 4 }
```

4.3.2.12.2.5 调度表日期值

```

date OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION

```

“32位整型数。当bit0为1时，表示该调度表1日需要执行，以此类推，bit30表示31日，bit31保留。”

```
 ::= { scheduleCfgEntry 5 }
```

4.3.2.12.2.6 调度表的日计划

```

dayPlan OBJECT-TYPE
    SYNTAX          INTEGER(1..128)
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION

```

“该调度表执行的日计划的编号。”

```
 ::= { scheduleCfgEntry 7 }
```

4.3.2.13 数据类13 运行状态

```
state OBJECT IDENTIFIER ::= { asc 13 }
```

-- 定义运行状态信息节点

-- 此节点中包含与信号控制机运行状态信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.运行状态(1.3.6.1.4.1.61332.3.2.2.13)。

4.3.2.13.1 设备状态

```
unitState OBJECT IDENTIFIER
 ::= { state 1 }
```

-- 定义设备状态信息节点

-- 此节点中包含与信号控制机设备状态相关的所有数据对象

4.3.2.13.1.1 检测器状态

detectorState OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(128))

ACCESS read-only

STATUS mandatory

DESCRIPTION

“当采用RAW格式时，检测器状态是一个16字节的数组，数组中的每个字节类型代表8个检测器，Byte Array [15]、...Byte Array [0]分别代表 bit127~bit120、...bit7~bit0，相应的bit位为1时，表示该检测器故障，为0时，表示该检测器正常，为1时表示该检测器故障。默认对应截面型车辆检测器。

当采用JSON格式时，检测器状态是一个包含128个整型数字的数组，数组中每个数字代表对应截面型车辆检测器的状态，数字为1代表对应检测器故障，数字为0代表对应检测器正常。”

::= { unitState 1 }

4.3.2.13.1.2 设备模块状态

moduleState OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(64))

ACCESS read-only

STATUS optional

DESCRIPTION

“当采用RAW格式时，设备模块状态是一个8字节的数组，数组中的每个字节类型代表8个模块，Byte Array[7]、...Byte Array [0]分别代表 bit63~bit56、...bit7~bit0，相应的bit位为1时，表示该模块故障，为0时，表示该模块正常，bit0~bit9表示主控板；bit10~bit39 表示相位板（bit10表示相位板1，以此类推）；bit40~bit59表示检测器板（bit40表示检测器板1，以此类推），其余bit保留。

当采用JSON格式时，设备模块状态是一个含64个整型数字的数组，每个数字对应一个模块的状态，数字为1代表对应模块故障，数字为0代表对应模块正常，模块对应关系参见RAW格式定义。”

::= { unitState 2 }

4.3.2.13.1.3 机柜门状态

cabinetDoor OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-only

STATUS optional

DESCRIPTION

“表示机柜各个门的开启状态，对应关系如下：

Bit0表示机柜前门状态；

Bit1表示机柜后门状态；

Bit2表示左侧门状态；

bit3表示右侧门状态；

其他保留。

所对应位为0：开门；1：关门”

::= { unitState 3 }

4.3.2.13.1.4 电压

voltage OBJECT-TYPE

SYNTAX INTEGER(0..512)

ACCESS read-only

STATUS optional

DESCRIPTION

“信号机工作的市电电压值，单位：v。”

::= { unitState 4 }

4.3.2.13.1.5 电流

```

current OBJECT-TYPE
    SYNTAX          INTEGER(0..65535)
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "信号机工作时使用的市电电流值，单位：0.1A。"
 ::= { unitState 5 }

```

4.3.2.13.1.6 温度

```

temp OBJECT-TYPE
    SYNTAX          INTEGER(-40..120)
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "信号机机柜内部的温度，单位：℃。"
 ::= { unitState 6 }

```

4.3.2.13.1.7 湿度

```

humidity OBJECT-TYPE
    SYNTAX          INTEGER(0..100)
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "信号机机柜内部的相对湿度。"
 ::= { unitState 7 }

```

4.3.2.13.1.8 水浸

```

water OBJECT-TYPE
    SYNTAX          INTEGER(0,1)
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "信号机机柜内部是否进水，0：未进水；1：进水。"
 ::= { unitState 8 }

```

4.3.2.13.1.9 烟雾

```

smoke OBJECT-TYPE
    SYNTAX          INTEGER(0,1)
    ACCESS           read-only
    STATUS           optional
    DESCRIPTION
        "信号机机柜内部是否有烟雾，0：没有烟雾；1：有烟雾。"
 ::= { unitState 9 }

```

4.3.2.13.1.10 信号机标准时间

```

UTCtime OBJECT-TYPE
    SYNTAX          Counter64(0..999999999999)
    ACCESS           read-write
    STATUS           mandatory
    DESCRIPTION
        "信号机当前的 UTC 时间戳。相对1970年1月1日0时0分0UTC起的毫秒计时，
        13位的十进制数。
        当采用RAW格式时，时间为8字节（Int64），高字节在前，低字节在后。
        当采用JSON格式时，时间为13位的十进制整数。"
 ::= { unitState 10 }

```

4.3.2.13.1.11 信号机本地时间

```

localTime OBJECT-TYPE
    SYNTAX          Counter64(0..999999999999)
    ACCESS           read-write
    STATUS           mandatory

```

DESCRIPTION

“信号机本地时间是信号机标准时间加上所属时区的时间差值。”

```
::= { unitState 11 }
```

4.3.2.13.2 运行状态建议分别列入其他数据类状态，与此前数据类状态合并

```
runState OBJECT IDENTIFIER
```

```
::= { state 2 }
```

-- 定义运行状态信息节点

-- 此节点中包含与信号控制机运行状态相关的所有数据对象

4.3.2.13.2.1 当前运行模式

```
curRunMode OBJECT-TYPE
```

```
SYNTAX INTEGER
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

“当前运行模式，当上位机查询时，代表路口当前的运行模式；当发生路口运行至周期切换，或者当信号机模式转换时，或者本地手动控制阶段改变时（控制模式不变，阶段号变化）或者中央阶段锁定开始、结束执行时，代表切换后的运行模式，应当主动上报。

值定义详见附录A.4运行模式相关。”

```
::= { runState 1 }
```

4.3.2.13.2.2 当前交通语义

```
curSemantic OBJECT-TYPE
```

```
SYNTAX INTEGER(
    0 (Semantic_PhaseStage),
    1 (Semantic_PhaseRing))
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

“当前运行交通语义，0代表阶段语义，1代表环语义，其他值非法。”

```
::= { runState 2 }
```

4.3.2.13.2.3 当前阶段方案

```
curStagePattern OBJECT-TYPE
```

```
SYNTAX INTEGER(0..128)
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

“当上位机查询时，代表路口当前的阶段方案编号；当发生路口运行至周期切换，或者发生信号机模式转换时，或者本地手动控制阶段改变时（控制模式不变，阶段号变化）或者中央阶段锁定开始、结束执行时，代表切换后的方案编号，应当在切换完成后主动上报。

当采用阶段语义时，方案对应阶段方案；当采用环语义时，本标识号无意义，应当返回0。”

```
::={ runState 3 }
```

4.3.2.13.2.4 当前阶段

```
curStage OBJECT-TYPE
```

```
SYNTAX INTEGER(0..64)
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

“当上位机查询时，代表路口当前的阶段编号；当发生路口运行至阶段切换，或者当发生信号机模式转换时，或者本地手动控制阶段改变时（控制模式不变，阶段号变化）或者中央阶段锁定开始、结束执行时，代表切换后的阶段，应当在切换后主动上报。

当采用阶段语义时，对应阶段编号；当采用环语义时，本标识号无意义，应当返回0。”

```
::= { runState 4 }
```

4.3.2.13.2.5 当前阶段时长

curStageTime OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(3))

ACCESS read-only

STATUS mandatory

DESCRIPTION

“包含三个整型数据的数组，分别对应阶段总时长、已运行时长、剩余时长，单位：1s。

当发生路口运行至阶段切换，或者当前运行阶段时长发生变化，应当主动上报。

当采用环语义时，本标识号无意义，应当返回全0。”

```
::= { runState 5 }
```

4.3.2.13.2.6 当前环方案

curRingPattern OBJECT-TYPE

SYNTAX INTEGER(0..128)

ACCESS read-only

STATUS optional

DESCRIPTION

“当上位机查询时，代表路口当前的环方案编号；当发生路口运行至周期切换，或者发生信号机模式转换时，或者本地手动控制阶段改变时（控制模式不变，阶段号变化）或者中央阶段锁定开始、结束执行时，代表切换后的方案编号，应当在切换完成后主动上报。

当采用环语义时，方案对应环方案；当采用阶段语义时，本标识号无意义，应当返回0。”

```
::= { runState 6 }
```

4.3.2.13.2.7 当前环相位

curPhase OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..8))

ACCESS read-only

STATUS optional

DESCRIPTION

“当上位机查询时，代表路口当前的环相位；当发生路口运行至环相位切换，或者当发生信号机模式转换时，或者本地手动控制阶段改变时（控制模式不变，阶段号变化）或者中央阶段锁定开始、结束执行时，代表切换后的环相位，应当在变化后主动上报。

当采用环语义时，对应环相位编号，最大8个环，每个字节对应一个环的相位编号。当采用阶段语义时，本标识号无意义，应当返回全0。”

```
::= { runState 7 }
```

4.3.2.13.2.8 当前环相位时长

curPhaseTime OBJECT-TYPE

SYNTAX SEQUENCE OF (SEQUENCE OF INTEGER[3])
(SIZE(1..8))

ACCESS read-only

STATUS optional

DESCRIPTION

“当上位机查询时，代表路口当前的环相位时长；

对于环语义，当发生多环中的任意一个相位切换，或者当前运行环相位时长发生变化时，应当主动上报。

该对象是一个二维数组，较高维对应环数量，最大8个环；较低维对应该环当前的相位时长数据，一共三个元素，对应相位总长、已运行时长、剩余时长。单位：1s。”

```
::= { runState 8 }
```

4.3.2.13.2.9 当前周期长度

```
curCycle OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "信号机运行的周期长度，单位：1s。
        当上位机查询时，代表路口当前的周期长度；
        当发生路口运行至周期切换，代表路口计划的周期长度；
        当控制方式为固定周期（Fixed-Time）时，该值恒定；
        当控制方式为感应控制（Actuated）、自适应控制（Adaptive）等具有周期
        动态调整能力的模式时，该值可能随周期实时变化，应结合控制模式一起解释
        使用；
        在感应或自适应控制下，该值反映上一周期或当前有效的调度周期，主要用于
        记录与分析。
        或者周期运行中发生周期变化时，应当主动上报。"
```

```
::= { runState 9 }
```

4.3.2.13.2.10 当前周期已运行时间

```
elapsedTime OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "上报时刻信号机本周期已运行的时间，单位：1s。
        当上位机查询时，代表路口当前的已运行时长；
        当发生报文主动推送，代表信号机在生成报文的时间周期已运行时长。"
```

```
::= { runState 10 }
```

4.3.2.13.2.11 协调状态信息

```
coordination OBJECT-TYPE
    SYNTAX          INTEGER(0,1)
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "信号机协调状态，0x01 为协调到位，0x00 为协调未到位。"
```

```
::= { runState 11 }
```

4.3.2.13.2.12 当前时间

```
curTime OBJECT-TYPE
    SYNTAX          Counter64(0..9999999999999)
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        " 信号机当前 UTC 时间戳，相对1970年1月1日0时0分0毫秒的UTC毫秒计时，
        13位的十进制数。
        当采用RAW格式时，时间为8字节（Int64），高字节在前，低字节在后。
        当采用JSON格式时，时间为13位的十进制整数。
        当上位机查询时，代表信号机当前时间；
        当发生报文主动推送，代表信号机生成报文的时间，应当随报文同步主动上报。
        "
```

```
::= { runState 12 }
```

4.3.2.14 数据类14 交通数据

```
trafficData OBJECT IDENTIFIER ::= { asc 14 }
-- 定义交通数据信息节点
-- 此节点中包含与信号控制机交通数据信息相关的所有数据对象
```

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.交通数据(1.3.6.1.4.1.61332.3.2.2.14)。

4.3.2.14.1 实时数据

detectorState OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(size(0..128))

ACCESS read-only

STATUS mandatory

DESCRIPTION

“该数据对应的是截面型车辆检测器整体实时状态。

当采用RAW格式时，实时数据是含16个字节的数组，数组中的每个字节类型代表8个截面检测器，Byte Array[15]、...Byte Array[0]分别代表bit127~bit120、...bit7~bit0，相应的 bit位为1时，表示该检测器截面上存在车辆，为0时，表示该检测器截面上不存在车辆。

当采用JSON格式时，实时数据是含128个整型数字的数组，数组中每个数字对应一个截面检测器状态，数字为1，表示检测器截面上存在车辆，为0时，表示该检测器截面上不存在车辆。”

::= { trafficData 1 }

4.3.2.14.2 检测器统计数据表

statisticTable OBJECT-TYPE

SYNTAX SEQUENCE OF statisticEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“包含每个检测器统计数据的信息数据组，数组元素数量等于对象标识

(4.3.2.)5.1 maxDetector。”

::= { trafficData 2 }

statisticEntry OBJECT-TYPE

SYNTAX statisticEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

“每个检测器统计数据的信息数据组。这是一个数据序列。”

INDEX { index }

::= { statisticTable 1 }

statisticEntry ::= SEQUENCE{

index INTEGER,

flow SEQUENCE OF INTEGER,

occupancy INTEGER,

avgSpeed INTEGER,

avgTimeHeadway INTEGER,

avgTimeGap INTEGER,

avgStopTimes INTEGER,

avgQueneLength INTEGER,

maxQueneLength INTEGER,

avgPedWaitNum INTEGER,

avgPedWaitTime INTEGER,

maxPedWaitNum INTEGER,

maxPedWaitTime INTEGER,

startTime Counter64,

endTime Counter64)

4.3.2.14.2.1 检测器编号

index OBJECT-TYPE

SYNTAX INTEGER(1..128)

ACCESS read-only

STATUS mandatory

DESCRIPTION

“信号机中检测器的序号，检测器1的编号为1，以此类推。”

::= { statisticEntry 1 }

4.3.2.14.2.2 流量

flow OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(4))

ACCESS read-only

STATUS optional

DESCRIPTION

“如果是机动车检测器，数据组成如下：

数据1：流量总数，Unsigned int16类型，2字节，范围0～65535；

数据2：大型车流量总数，Unsigned int16类型，2字节，范围0～65535；

数据3：中型车流量总数，Unsigned int16类型，2字节，范围0～65535；

数据4：小型车流量总数，Unsigned int16类型，2字节，范围0～65535。

车辆类型区分标准如下：

大型车：车长 ≥ 12 m；

中型车：6 m $\leq$ 车长 < 12 m；

小型车：车长 < 6 m

如果是非机动车检测器，数据组成如下：

数据1：流量总数，Unsigned int16类型，2字节，范围0～65535；

数据2：自行车流量总数，Unsigned int16类型，2字节，范围0～65535；

数据3：电动自行车流量总数，Unsigned int16类型，2字节，范围0～65535；

数据4：其他非机动车流量总数，Unsigned int16类型，2字节，范围0～65535。

如果是行人检测器，数据组成如下：

数据1：人行横道总人流量，2字节，范围0～65535，单位：1人；

数据2：人行横道上行人流量，2字节，范围0～1000，单位：1人；

数据3：人行横道下行人流量，2字节，范围0～1000，单位：1人；

数据4：无意义，填0。

当采用RAW格式时，该对象共8字节，每两个字节对应一个数据，每个数据高字节在前，低字节在后。

当采用JSON格式时，是一个包含4个整型数字的数组，每个数字对应一项数据。”

::= { statisticEntry 2 }

4.3.2.14.2.3 平均占有率

occupancy OBJECT-TYPE

SYNTAX INTEGER(0..200)

ACCESS read-only

STATUS optional

DESCRIPTION

“信号机检测器上采集周期内的时间占有率值，单位：0.5%”

::= { statisticEntry 3 }

4.3.2.14.2.4 平均速度

avgSpeed OBJECT-TYPE

SYNTAX INTEGER(0..300)

ACCESS read-only

STATUS optional

DESCRIPTION

“信号机检测器上统计周期内通过的车辆速度的平均值，单位：km/h。”

::= { statisticEntry 4 }

4.3.2.14.2.5 平均车头时距

avgTimeHeadway OBJECT-TYPE

SYNTAX INTEGER(0..65535)

ACCESS read-only
STATUS optional
DESCRIPTION

“检测截面过车平均车头时距，整型数据，2字节，单位：0.1秒。”

::= { statisticEntry 5 }

4.3.2.14.2.6 平均车间时距

avgTimeGap OBJECT-TYPE
SYNTAX INTEGER(0..65535)
ACCESS read-only
STATUS optional
DESCRIPTION

“检测截面过车平均车间时距，整型数据，2字节，单位：0.1秒。”

::= { statisticEntry 6 }

4.3.2.14.2.7 平均停车次数

avgStopTimes OBJECT-TYPE
SYNTAX INTEGER(0..255)
ACCESS read-only
STATUS optional
DESCRIPTION

“检测截面过车平均停车次数，整型数据，1字节，单位：0.1次。”

::= { statisticEntry 7 }

4.3.2.14.2.8 平均排队长度

avgQueueLength OBJECT-TYPE
SYNTAX INTEGER(0..65535)
ACCESS read-only
STATUS optional
DESCRIPTION

“车道采样后期内排队长度数据的平均值，2字节，单位：1米。”

::= { statisticEntry 8 }

4.3.2.14.2.9 最大排队长度

maxQueueLength OBJECT-TYPE
SYNTAX INTEGER(0..65535)
ACCESS read-only
STATUS optional
DESCRIPTION

“车道采样周期内达到的最大排队长度，2字节，单位：1米。”

::= { statisticEntry 9 }

4.3.2.14.2.10 等候区平均等候人数

avgPedWaitNum OBJECT-TYPE
SYNTAX INTEGER(0..65535)
ACCESS read-only
STATUS optional
DESCRIPTION

“区域采样周期内等候人数采样数据的平均值，2字节，单位：0.1人。”

::= { statisticEntry 10 }

4.3.2.14.2.11 等候区平均等候时间

avgPedWaitTime OBJECT-TYPE
SYNTAX INTEGER(0..65535)
ACCESS read-only
STATUS optional
DESCRIPTION

“区域采样周期内平均等候时间采样数据的平均值，2字节，单位：0.1秒；”

::= { statisticEntry 11 }

4.3.2.14.2.12 等候区最大等候人数

maxPedWaitNum OBJECT-TYPE

```

SYNTAX          INTEGER(0..65535)
ACCESS          read-only
STATUS          optional
DESCRIPTION

```

“区域采样周期内达到的等候人数最大值，2字节，单位：1人。”

```
 ::= { statisticEntry 12 }
```

4.3.2.14.2.13 等候区最大等候时间

```

maxPedWaitTime OBJECT-TYPE
    SYNTAX          INTEGER(0..65535)
    ACCESS          read-only
    STATUS          optional
    DESCRIPTION

```

“区域采样周期内个体最大等候时间，2字节，单位：1秒；”

```
 ::= { statisticEntry 13 }
```

4.3.2.14.2.14 统计开始时间

```

startTime OBJECT-TYPE
    SYNTAX          Counter64(0..9999999999999)
    ACCESS          read-only
    STATUS          optional
    DESCRIPTION

```

“交通数据统计的起始时间，单位：毫秒。

相对1970年1月1日0时0分0毫秒的毫秒计时，13位的十进制数。

当采用RAW格式时，时间为8字节（Int64），高字节在前，低字节在后。

当采用JSON格式时，时间为13位的整形数据。”

```
 ::= { statisticEntry 14 }
```

4.3.2.14.2.15 统计结束时间

```

endTime OBJECT-TYPE
    SYNTAX          Counter64(0..9999999999999)
    ACCESS          read-only
    STATUS          optional
    DESCRIPTION

```

“交通数据统计的结束时间，单位：毫秒。

相对1970年1月1日0时0分0毫秒的毫秒计时，13位的十进制数。

当采用RAW格式时，时间为8字节（Int64），高字节在前，低字节在后。

当采用JSON格式时，时间为13位的整形数据。”

```
 ::= { statisticEntry 15 }
```

4.3.2.15 数据类15 报警数据

```
alarm OBJECT IDENTIFIER ::= { asc 15 }
```

-- 定义报警数据信息节点

-- 此节点中包含与信号控制机报警数据信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.报警数据(1.3.6.1.4.1.61332.3.2.2.15)。

4.3.2.15.1 当前报警数量

```

maxAlarm OBJECT-TYPE
    SYNTAX          INTEGER(0..65535)
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION

```

“信号机内当前已经使用的最大的报警编号。”

```
 ::= { alarm 1 }
```

4.3.2.15.2 报警数据表

```
alarmTable OBJECT-TYPE
```

SYNTAX SEQUENCE OF alarmEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION

“包含每个报警信息的信息数据组，数组元素数量等于maxAlarm。”

::= { alarm 2 }

alarmEntry OBJECT-TYPE

SYNTAX alarmEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION

“每个报警信息的信息数据组。这是一个数据序列。”

INDEX { index }

::= { alarmTable 1 }

alarmEntry ::= SEQUENCE{

index INTEGER,
 type INTEGER,
 value INTEGER,
 time Counter64,
 detail SEQUENCE OF INTEGER)

4.3.2.15.2.1 报警编号

index OBJECT-TYPE

SYNTAX INTEGER(1..65535)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION

“信号机中报警数据的序号，报警数据1编号为1，以此类推。”

::= { alarmEntry 1 }

4.3.2.15.2.2 报警类型

type OBJECT-TYPE

SYNTAX INTEGER(
 0x00 (Type_Alarm_Unavailable),
 0x10 (Type_Alarm_Light),
 0x30 (Type_Alarm_Detector),
 0x40 (Type_Alarm_Device),
 0x60 (Type_Alarm_Environment),
 0xF0 (Type_Alarm_Reboot),
 0xFF (Type_Alarm_Undefined))
 ACCESS read-only
 STATUS optional
 DESCRIPTION

“信号机报警类型定义如下：

- | | |
|--------------------------------------|-------|
| a) 无效报警 (Type_Alarm_Unavailable) | 0x00 |
| b) 信号灯报警 (Type_Alarm_Light) | 0x10 |
| c) 检测器报警 (Type_Alarm_Detector) | 0x30 |
| d) 设备故障报警 (Type_Alarm_Device) | 0x40 |
| e) 工作环境异常报警 (Type_Alarm_Environment) | 0x60 |
| f) 设备重启上报 (Type_Alarm_Reboot) | 0xF0 |
| g) 本标准未定义报警 (Type_Alarm_Undefined) | 0xFF” |

::= { alarmEntry 2 }

4.3.2.15.2.3 报警值

value OBJECT-TYPE

SYNTAX INTEGER(0..255)
 ACCESS read-only
 STATUS optional

DESCRIPTION

“信号机报警值定义如下：

- a) 信号灯报警类型的值(1~192)，表示64个通道中的192个灯头报警值，排列的顺序为：通道1的红、黄、绿，通道2的红、黄、绿…；
- b) 检测器报警的值为：1~128，表示具体编号的检测器报警；
- c) 设备报警的值为：1~255，表示具体设备发生报警：
 - 1) 1~10：控制板报警；
 - 2) 11~40：相位板报警，11表示相位板1报警，以此类推；
 - 3) 41~60：检测板报警，41表示检测板1报警，以此类推；
 - 4) 255：表示黄闪报警。
- d) 工作环境报警值定义：
 - 1) 0x10—电压异常，0x11—电压过高，0x12—电压过低，0x13—断电；
 - 2) 0x20—温度异常，0x21—温度过高，0x22—温度过低；
 - 3) 0x30—湿度异常，0x31—湿度过高，0x32—湿度过低；
 - 4) 0x40—烟雾报警；
 - 5) 0x50—震动报警；
 - 6) 0x60—水浸报警；
 - 7) 0x70—开关门报警，0x71—前门打开报警，0x72—后门打开报警，0x74—左侧门打开报警，0x78—右侧门打开报警。（注：左侧门一般指手动门，右侧门指应急电源门）
- e) 设备重启类型自定义：0 未知原因重启，1 中央平台下发重启指令。”

```
::= { alarmEntry 3 }
```

4.3.2.15.2.4 报警时间

time OBJECT-TYPE

SYNTAX Counter64 (0..99999999999999)

ACCESS read-only

STATUS optional

DESCRIPTION

“报警事件发生时间，单位：毫秒。

相对1970年1月1日0时0分0毫秒的毫秒计时，13位的十进制数。

当采用RAW格式时，时间为8字节（Int64），高字节在前，低字节在后。

当采用JSON格式时，时间为13位的整形数据。”

```
::= { alarmEntry 4 }
```

4.3.2.15.2.5 报警详细信息

detail OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(2)

ACCESS read-only

STATUS optional

DESCRIPTION

“描述报警类型报警值的详细信息。根据报警类型的不同，详细信息的内容也不同。

- a) 信号灯报警 Array[0]、Array[1]保留；
- b) 检测器报警 Array[0]、Array[1]保留；
- c) 设备故障报警 Array[0]、Array[1]保留；
- d) 工作环境异常报警：报警值不同，详细信息内容也不同，两字节表示的高位在前。
 - 1) Array[0]和Array[1]表示异常电压的值，断电为0，单位：伏；
 - 2) Array[0]和Array[1]表示异常温度的值，范围-40~120，单位℃；
 - 3) Array[0]和Array[1]表示异常湿度的值，范围0~100，单位1%；
 - 4) Array[0]和Array[1]表示烟雾浓度的值，范围0~100，单位1%；
 - 5) Array[0]、Array[1]保留；
 - 6) Array[0]、Array[1]保留；
 - 7) Array[0]、Array[1]保留；
- e) 溢出报警：Array[0]、Array[1]保留；

f) 设备重启: Array[0]表示重启的具体原因:

0: 无效的数据;
1: 代表远程指令重启;
2: 代表本地指令重启;
3: 代表断电重启;
4: 代表程序自行重启;
5: 代表程序崩溃重启;
255: 未定义的原因。
Array[1]保留。”

```
::= { alarmEntry 5 }
```

4.3.2.16 数据类16 故障数据

```
fault OBJECT IDENTIFIER ::= { asc 16 }
```

-- 定义故障数据信息节点

-- 此节点中包含与信号控制机故障数据信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时, 设备标识号路径为:

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.故障数据(1.3.6.1.4.1.61332.3.2.2.16)。

4.3.2.16.1 当前故障数量

```
maxFault OBJECT-TYPE
```

```
SYNTAX          INTEGER(0..65535)
```

```
ACCESS          read-only
```

```
STATUS          mandatory
```

```
DESCRIPTION
```

“信号机内当前已经使用的最大的故障编号。”

```
::= { fault 1 }
```

4.3.2.16.2 故障记录表

```
faultTable OBJECT-TYPE
```

```
SYNTAX          SEQUENCE OF faultEntry
```

```
ACCESS          not-accessible
```

```
STATUS          mandatory
```

```
DESCRIPTION
```

“包含每个故障信息的信息数据组, 数组元素数量等于maxFault。”

```
::= { fault 2 }
```

```
faultEntry OBJECT-TYPE
```

```
SYNTAX          faultEntry
```

```
ACCESS          not-accessible
```

```
STATUS          mandatory
```

```
DESCRIPTION
```

“每个故障信息的信息数据组。这是一个数据序列。”

```
INDEX          {index}
```

```
::= { faultTable 1 }
```

```
faultEntry ::= SEQUENCE{
```

```
    index          INTEGER,
```

```
    type           INTEGER,
```

```
    time           Counter64,
```

```
    action         INTEGER,
```

```
    detail         SEQUENCE OF INTEGER)
```

4.3.2.16.2.1 故障编号

```
index OBJECT-TYPE
```

```
SYNTAX          INTEGER(1..65535)
```

```
ACCESS          read-only
```

```
STATUS          mandatory
```

DESCRIPTION

“信号机中故障数据的序号，故障数据1编号为1，以此类推。”

```
::= { faultEntry 1 }
```

4.3.2.16.2.2 故障类型

```
type OBJECT-TYPE
```

```
    SYNTAX          INTEGER(0..255)
```

```
    ACCESS          read-only
```

```
    STATUS          optional
```

```
    DESCRIPTION
```

“信号机故障类型定义如下：

a) 无效故障 (Type_Fault_Unavailable)	0x00
b) 绿冲突故障 (Type_Fault_GreenConflict)	0x10
c) 红绿冲突故障 (Type_Fault_GreenRedConflict)	0x11
d) 红灯全灭故障 (Type_Fault_RedLightAllOff)	0x12
e) 红灯故障 (Type_Fault_RedLight)	0x20
f) 黄灯故障 (Type_Fault_YellowLight)	0x21
g) 绿灯故障 (Type_Fault_GreenLight)	0x22
h) 通信故障 (Type_Fault_Communication)	0x30
i) 自检故障 (Type_Fault_Self)	0x40
j) 检测器故障 (Type_Fault_Detector)	0x41
k) 继电器故障 (Type_Fault_Relay)	0x42
l) 存储器故障 (Type_Fault_Memory)	0x43
m) 时钟故障 (Type_Fault_Clock)	0x44
n) 主板故障 (Type_Fault_MontherBoard)	0x45
o) 相位板故障 (Type_Fault_PhaseBoard)	0x46
p) 检测板故障 (Type_Fault_DetetorBoard)	0x47
q) IO板故障 (Type_Fault_IOBoard)	0x48
r) 黄闪器故障 (Type_Fault_YellowFlash)	0x49
s) 总线故障 (Type_Fault_Bus)	0x4A
t) 相位板插头编码故障 (Type_Fault_PhasePlugID)	0x4B
u) 检测板插头编码故障 (Type_Fault_DetectorPlugID)	0x4C
v) IO板插头编码故障 (Type_Fault_IOPlugID)	0x4D
w) 配置故障 (Type_Fault_Config)	0x50
x) 特征参数丢失故障 (Type_Fault_PersonalityLose)	0x60
y) 特征参数格式错误 (Type_Fault_PeronalityFormat)	0x61
z) 特征参数与地址板匹配故障 (Type_Fault_PersonalityID_UnMatch)	0x62
aa) 控制响应故障 (Type_Fault_Response)	0x70
ab) 本标准未涵盖的故障类型 (Type_Fault_Undefined)	0xFF”

```
::= { faultEntry 2 }
```

4.3.2.16.2.3 故障时间

```
time OBJECT-TYPE
```

```
    SYNTAX          Counter64(0..99999999999999)
```

```
    ACCESS          read-only
```

```
    STATUS          optional
```

```
    DESCRIPTION
```

“故障发生时间，单位：毫秒。

相对1970年1月1日0时0分0毫秒的毫秒计时，13位的十进制数。

当采用RAW格式时，时间为8字节（Int64），高字节在前，低字节在后。

当采用JSON格式时，时间为13位的整形数据。”

```
::= { faultEntry 3 }
```

4.3.2.16.2.4 故障动作

```
action OBJECT-TYPE
```

```
    SYNTAX          INTEGER(
                        0x00(Switch_Null),
```

```

                                0x10(Switch_To_Flash),
                                0x20(Switch_To_Off),
                                0x30(Switch_To_Red),
                                0x40(Switch_To_Local_FixCycle),
                                0x50(Switch_To_Local_Coordination),
                                0x60(Switch_To_Local_Va),
                                0xFF(Switch_Undefined))
ACCESS                          read-only
STATUS                          optional
DESCRIPTION
    “信号机故障动作定义如下：
    a) 无故障动作 (Switch_Null)                                0x00
    b) 切换到黄闪 (Switch_To_Flash)                            0x10
    c) 切换到灭灯 (Switch_To_Off)                              0x20
    d) 切换到全红 (Switch_To_Red)                              0x30
    e) 切换到本地定周期 (Switch_To_Local_FixCycle)            0x40
    f) 切换到本地协调 (Switch_To_Local_Coordination)          0x50
    g) 切换到本地感应 (Switch_To_Local_Va)                    0x60
    h) 未定义动作 (Switch_Undefined)                          0xFF”
    ::= { faultEntry 2 }

```

4.3.2.16.2.5 故障详细信息

```

detail OBJECT-TYPE
    SYNTAX          SEQUENCE OF INTEGER(SIZE(2))
    ACCESS          read-only
    STATUS          optional
    DESCRIPTION

```

“描述故障的详细信息。根据故障类型的不同，详细信息的内容也不同。

- a) 绿冲突故障：Array[0]、Array[1]表示发生绿冲突的灯组编号；
 - b) 红绿冲突故障：Array[0]表示发生红绿冲突的灯组编号，Array[1]为0表示故障，为1表示故障恢复；
 - c) 红灯故障：Array[0]表示发生红灯故障的灯组编号，Array[1]为0表示故障，为1表示故障恢复；
 - d) 黄灯故障：Array[0]表示发生黄灯故障的灯组编号，Array[1]为0表示故障，为1表示故障恢复；
 - e) 绿灯故障：Array[0]表示发生绿灯故障的灯组编号，Array[1]为0表示故障，为1表示故障恢复；
 - f) 通信故障：Array[0]、Array[1]保留；
 - g) 自检故障：Array[0]保留，Array[1]为0表示故障，为1表示故障恢复；
 - h) 检测器故障：Array[0]表示故障检测器的编号，Array[1]为0表示故障，为1表示故障恢复；
 - i) 继电器故障：Array[0]表示故障继电器的编号，Array[1]为0表示故障，为1表示故障恢复；
 - j) 存储器故障：Array[0]保留，Array[1]为0表示故障，为1表示故障恢复；
 - k) 时钟故障：Array[0]保留，Array[1]为0表示故障，为1表示故障恢复；
 - l) 主板故障：Array[0]保留，Array[1]为0表示故障，为1表示故障恢复；
 - m) 相位板故障：Array[0]表示故障相位板的编号，Array[1]为0表示故障，为1表示故障恢复；
 - n) 检测板故障：Array[0]表示故障检测板的编号，Array[1]为0表示故障，为1表示故障恢复；
 - o) 配置故障：Array[0]保留，Array[1]为0表示故障，为1表示故障恢复；
 - p) 控制响应故障：Array[0]保留，Array[1]为0表示故障，为1表示故障恢复。”
- ```

 ::= { faultEntry 5 }

```

#### 4.3.2.17 数据类17 中心控制

```
remoteControl OBJECT IDENTIFIER ::= { asc 17 }
-- 定义中心控制信息节点
-- 此节点中包含与信号控制机中心控制信息相关的所有数据对象
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
 iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
 设备.交通信号控制机.中心控制(1.3.6.1.4.1.61332.3.2.2.17)。
```

#### 4.3.2.17.1 中心控制命令

```
command OBJECT IDENTIFIER
::= { remoteControl 1 }
-- 定义中心控制命令节点
-- 此节点中包含与中心控制命令相关的所有数据对象
```

##### 4.3.2.17.1.1 指定状态

```
assignState OBJECT IDENTIFIER
::= { command 1 }
-- 定义指定状态命令节点
-- 此节点中包含与指定状态相关的所有数据对象
```

##### 4.3.2.17.1.1.1 指定阶段

```
assignStage OBJECT-TYPE
 SYNTAX INTEGER(1..64)
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "当信号机运行在阶段语义下时，为单字节数据，代表中心指定路口运行指定的
 阶段编号。
 当运行在环语义下时，返回控制失败错误。"
 ::= { assignState 1 }
```

##### 4.3.2.17.1.1.2 指定相位

```
assignPhase OBJECT-TYPE
 SYNTAX SEQUENCE OF INTEGER(SIZE(1..64))
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "当信号机运行在环语义下时，指定运行的相位编号列表。
 如果采用RAW格式，为8字节数据(64 bit)，bit0代表相位1，bit63代表相
 位64，数据结构为 BYTE8、...BYTE1，代表 bit63~bit56、...bit7~bit0，
 相应的 bit 位为1时，表示中心指定路口运行的相位包含该相位。
 如果采用JSON格式，为含1到64个整型数字的数组，每个数字代表一个相位的
 编号。
 当运行在阶段语义下时，返回控制失败错误。"
 ::= { assignState 2 }
```

##### 4.3.2.17.1.1.3 指定时长

```
assignTime OBJECT-TYPE
 SYNTAX INTEGER(0..255)
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "中心指定路口阶段或环锁定时，该字段值表示锁定的时长。0表示取消相应的
 中心控制，255表示一直锁定，直到发取消指令。"
 ::= { assignState 3 }
```

##### 4.3.2.17.1.1.4 指定模式

```
assignMode OBJECT-TYPE
 SYNTAX INTEGER(
 1(normalMode),
```

|             |                      |      |
|-------------|----------------------|------|
|             | 2 (quickMode),       |      |
|             | 3 (guardMode))       |      |
| ACCESS      | read-write           |      |
| STATUS      | optional             |      |
| DESCRIPTION | “中心指定锁定模式，包含以下选项：    |      |
|             | a) 正常锁定 (normalMode) | 0x01 |
|             | b) 快速锁定 (quickMode)  | 0x02 |
|             | c) 警卫锁定 (guardMode)  | 0x03 |

**注：**正常锁定模式下非锁定阶段正常放行；快速锁定模式下待锁定阶段前的阶段走最小绿后快速过渡到下个阶段，不改变相序；警卫锁定模式下则允许改变相序，调相过渡到待锁定阶段；对于环结构，根据相位进行锁定操作。”

::= { assignState 4 }

#### 4.3.2.17.1.2 指定方案

assignPattern OBJECT IDENTIFIER

::= { command 2 }

-- 定义指定方案命令节点

-- 此节点中包含与指定方案相关的所有数据对象

##### 4.3.2.17.1.2.1 指定阶段方案

assignStagePattern OBJECT-TYPE

|        |                 |
|--------|-----------------|
| SYNTAX | INTEGER(1..128) |
| ACCESS | read-write      |
| STATUS | mandatory       |

DESCRIPTION

“中心指定路口运行指定的阶段方案的编号。

若目前信号机实际运行的语义不是阶段语义，返回控制失败错误。”

::= { assignPattern 1 }

##### 4.3.2.17.1.2.2 指定环方案

assignRingPattern OBJECT-TYPE

|        |                 |
|--------|-----------------|
| SYNTAX | INTEGER(1..128) |
| ACCESS | read-write      |
| STATUS | optional        |

DESCRIPTION

“中心指定路口运行指定的环方案的编号。

若目前信号机实际运行的语义不是环语义，返回控制失败错误。”

::= { assignPattern 2 }

##### 4.3.2.17.1.2.3 指定运行周期数

assignCycleTimes OBJECT-TYPE

|        |                 |
|--------|-----------------|
| SYNTAX | INTEGER(0..255) |
| ACCESS | read-write      |
| STATUS | mandatory       |

DESCRIPTION

“中心指定路口方案时，该字段值表示该方案执行的周期次数。

0表示取消相应的中心控制，255表示一直锁定，直到发取消指令。”

::= { assignPattern 3 }

##### 4.3.2.17.1.2.4 指定预计生效时间

assignActionDelay OBJECT-TYPE

|        |                 |
|--------|-----------------|
| SYNTAX | INTEGER(0..255) |
| ACCESS | read-write      |
| STATUS | mandatory       |

DESCRIPTION

“中心指定路口方案时，该字段值表示该方案从收到指令时刻开始，预计多长时间以后开始执行。为0代表立即执行；不为0，则信号机应当根据自身运行状况调整，确保指定的时间到达时开始立即执行指定控制方案。”

::= { assignPattern 4 }

## 4.3.2.17.1.2.5 指定模式

```

assignMode OBJECT-TYPE
 SYNTAX INTEGER(0..255)
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "当指定方案时，该字段值定义详见附录A.4运行模式。"
 ::= { assignPattern 5 }

```

## 4.3.2.17.1.3 方向锁定配置表

```

dirLockCfgTable OBJECT-TYPE
 SYNTAX SEQUENCE OF dirLockCfgEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "包含每个方向锁定信息的信息数据组，数组元素数量等于8。"
 ::= { command 3 }

```

```

dirLockCfgEntry OBJECT-TYPE
 SYNTAX dirLockCfgEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "每个方向锁定信息的信息数据组。这是一个数据序列。"
 INDEX { dirIndex }
 ::= { dirLockCfgTable 1 }

```

```

dirLockCfgEntry ::= SEQUENCE{
 dirIndex INTEGER,
 dirLockStage INTEGER,
 dirLockPhaseChain SEQUENCE OF INTEGER}

```

## 4.3.2.17.1.3.1 方向序号

```

dirIndex OBJECT-TYPE
 SYNTAX INTEGER(
 1 (Dir_East),
 2 (Dir_West),
 3 (Dir_South),
 4 (Dir_North),
 5 (Dir_EastWest_Straight),
 6 (Dir_EastWest_Left),
 7 (Dir_SouthNorth_Straight),
 8 (Dir_SouthNorth_Left))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "方向定义如下：
 1:东向单放 (Dir_East);
 2:西向单放 (Dir_West);
 3:南向单放 (Dir_South);
 4:北向单放 (Dir_North);
 5:东西直行对放 (Dir_EastWest_Straight);
 6:东西左转对放 (Dir_EastWest_Left);
 7:南北直行对放 (Dir_SouthNorth_Straight);
 8:南北左转对放 (Dir_SouthNorth_Left)。
 其他数值均为无效配置。"
 ::= { dirLockCfgEntry 1 }

```

## 4.3.2.17.1.3.2 方向锁定阶段

```

dirLockStage OBJECT-TYPE

```

SYNTAX INTEGER(1..64)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION

“信号机手控方向锁定对应的阶段。”

::= { dirLockCfgEntry 2 }

#### 4.3.2.17.1.3.3 方向锁定环相位链

dirLockPhaseChain OBJECT-TYPE  
 SYNTAX SEQUENCE OF INTEGER(SIZE(1..8))  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION

“信号机手控方向锁定对应的相位序列。”

::= { dirLockCfgEntry 3 }

#### 4.3.2.17.2 协调控制信息

coordination OBJECT IDENTIFIER  
 ::= { remoteControl 2 }

-- 定义中心协调控制命令节点

-- 此节点中包含与中心协调控制命令相关的所有数据对象

##### 4.3.2.17.2.1 协调子区周期

cycle OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 “协调子区周期，单位： 秒。”  
 ::= { coordination 1 }

##### 4.3.2.17.2.2 协调相位差

offset OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 “协调相位差，单位： 秒。”  
 ::= { coordination 2 }

##### 4.3.2.17.2.3 相位差调节需求

offsetAdjRequest OBJECT-TYPE  
 SYNTAX INTEGER(  
 0 (OffsetAdj\_OFF),  
 1 (OffsetAdj\_ON),  
 2 (Coordination\_OFF))  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 “调节需求含义如下：  
 值为 0：表示信号机不进行相位差调节（心跳维护）(OffsetAdj\_OFF)；  
 值为 1：表示请求信号机进行相位差调节 (OffsetAdj\_ON)；  
 值为 2：表示取消协调 (Coordination\_OFF)。”  
 ::= { coordination 3 }

##### 4.3.2.17.2.4 调整周期数

adjCycleTimes OBJECT-TYPE  
 SYNTAX INTEGER(1..4)  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION

“信号机相位差调整的最大周期数，默认为 2 周期调整到位。”  
 ::= { coordination 4 }

#### 4.3.2.17.2.5 协调绿信比模式

splitMode OBJECT-TYPE  
 SYNTAX INTEGER(  
     0(Split\_LocalAssign),  
     1(Split\_RemoteAssign))  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
     “绿信比模式含义如下：  
     值为 0：表示中心不指定绿信比参数，信号机可以自行分配时长，但应当保证  
     周期满足协调要求(Split\_LocalAssign)。  
     值为 1：表示中心指定协调绿信比参数，信号机根据协调绿信比参数中给定的  
     时间执行(Split\_RemoteAssign)。”  
 ::= { coordination 5 }

#### 4.3.2.17.2.6 协调阶段控制

stageControl OBJECT IDENTIFIER  
 ::= { coordination 6 }  
 -- 定义中心协调阶段控制命令节点  
 -- 此节点中包含与中心协调阶段控制命令相关的所有数据对象

##### 4.3.2.17.2.6.1 协调阶段

coordinationStage OBJECT-TYPE  
 SYNTAX INTEGER(1..64)  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
     “需要协调的阶段编号。”  
 ::= { stageControl 1 }

##### 4.3.2.17.2.6.2 指定协调阶段方案

coordinationPattern OBJECT-TYPE  
 SYNTAX INTEGER(0..128)  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
     “信号机协调时指定具体方案（1~128），表示信号机应当切换该指定的方案，  
     并且只在此方案进行协调，不根据日计划时段进行方案切换（用于警卫绿波协  
     调）；普通日计划时段协调，该字段填0。  
     若目前信号机实际运行的语义非阶段语义，返回控制失败错误。”  
 ::= { stageControl 2 }

##### 4.3.2.17.2.6.3 协调阶段绿信比参数

coordinationSplit OBJECT-TYPE  
 SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
     “当采用RAW格式时：  
     每两个字节代表一个阶段的时间，高字节在前，低字节在后，一共16个阶段，  
     总长32字节。  
     当采用JSON格式时：  
     该属性是一个包含1到16个整型数据的数组，每个数据代表一个阶段的时间，  
     数组中元素的数量应该和方案的阶段数量匹配。  
     若目前信号机实际运行的语义不是阶段语义，返回控制失败错误。”  
 ::= { stageControl 3 }

## 4.3.2.17.2.7 协调环控制

```
ringControl OBJECT IDENTIFIER
 ::= { coordination 7 }
-- 定义中心协调环控制命令节点
-- 此节点中包含与中心协调环控制命令相关的所有数据对象
```

## 4.3.2.17.2.7.1 协调相位

```
coordinationPhase OBJECT-TYPE
 SYNTAX INTEGER(1..64)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "需要协调的相位编号。"
 ::= { ringControl 1 }
```

## 4.3.2.17.2.7.2 指定协调环方案

```
coordinationPattern OBJECT-TYPE
 SYNTAX INTEGER(0..128)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "信号机协调时指定具体方案（1~128），表示信号机应当切换该指定的方案，
 并且只在此方案进行协调，不根据日计划时段进行方案切换（用于警卫绿波协调）；
 普通日计划时段协调，该字段填 0。
 若目前信号机实际运行的语义非环语义，返回控制失败错误。"
 ::= { ringControl 2 }
```

## 4.3.2.17.2.7.3 协调环绿信比参数

```
coordinationSplit OBJECT-TYPE
 SYNTAX SEQUENCE OF (SEQUENCE OF INTEGER(SIZE(2..16)))
 (SIZE(1..8))
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "信号机协调指定环绿信比参数，二维数组，整型数据，单位：1s。
 当采用RAW格式时：
 固定8个环，对于每个环中的每个相位绿信比时间，用连续两个字节代表，高字节在前，低字节在后，Array[0][[0]~Array[0][31]代表方案中环1总共16个相位的绿信比时间。总体是一个8*32的数组。
 当采用JSON格式时：
 该属性是一个二维数组，第一维对应环的数量，最大允许8个环，第二维是一个包含1到16个整型数字的数组，每个数字对应环方案中对应相位的绿信比时间。
 若目前信号机实际运行的语义不是环语义，返回控制失败错误。"
 ::= { ringControl 3 }
```

## 4.3.2.17.3 中心临时方案

```
tempPattern OBJECT IDENTIFIER
 ::= { remoteControl 3 }
-- 定义中心临时方案命令节点
-- 此节点中包含与中心临时方案控制命令相关的所有数据对象
```

## 4.3.2.17.3.1 方案运行模式

```
runMode OBJECT-TYPE
 SYNTAX INTEGER(0..255)
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "当指定方案时，代表方案运行模式，该字段值定义详见附录A.4运行模式。"
 ::= { tempPattern 1 }
```

## 4.3.2.17.3.2 方案相位差时间

offset OBJECT-TYPE

SYNTAX INTEGER(0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“相位差时间，单位：1s。

中心在制定临时方案时，可以更改继承方案的相位差时间。

注意如果当前运行的方案和更改后的相位差时间不一致，可能要经过多个周期的调整才能达成。平台需要通过运行状态表的协调状态信息((4.3.2.)13.2.11)来获取，同时指定的临时方案运行周期数需要保证能够完成相位差调整时间。”

::= { tempPattern 2 }

## 4.3.2.17.3.3 方案相位差计算参考点

offsetSyncPoint OBJECT-TYPE

SYNTAX INTEGER(  
0(Offset\_Ref\_Unavailable),  
1(Offset\_Ref\_GreenOn),  
2(Offset\_Ref\_GreenOff),  
3(Offset\_Ref\_YellowOff))

ACCESS read-write

STATUS mandatory

DESCRIPTION

“中心在制定临时方案时，可以根据需要更改协调相位差计算参考点。

相位差计算参考点类型定义如下：

- |                                      |      |
|--------------------------------------|------|
| a) 未设置 (Offset_Ref_Unavailable)      | 0x00 |
| b) 相对协调相位绿灯开始 (Offset_Ref_GreenOn)   | 0x01 |
| c) 相对协调相位绿灯结束 (Offset_Ref_GreenOff)  | 0x02 |
| d) 相对协调相位黄灯结束 (Offset_Ref_YellowOff) | 0x03 |

**注：**本标准参考GB/T 20999-2017，对于相位过渡的定义为“关闭、启动路权过渡灯色1、2、3”，逻辑严密的角度，应当将此类型定义为相对“关闭、启动路权过渡灯色1、2、3”，但考虑到此种方式难于理解并且不符合一般习惯，因此仍然定义为符合日常习惯的相对灯色的定义。”

::= { tempPattern 3 }

## 4.3.2.17.3.4 临时方案运行周期数

totalCycleTimes OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

“当前临时方案运行总周期数量。当要求切换到中心临时方案，而此参数设置为0，切换失败，返回参数错误(Status\_BadValue)；

中心如果希望提前结束临时方案，可将该数值置为0，则信号机在当前周期完成后自动退出中心控制临时方案。”

::= { tempPattern 4 }

## 4.3.2.17.3.5 临时方案执行状态

runState OBJECT-TYPE

SYNTAX INTEGER(  
0(State\_Wait),  
1(State\_Run))

ACCESS read-only

STATUS mandatory

DESCRIPTION

“当前临时方案运行状态。

0:未执行(State\_Wait)；

1: 执行中(State\_Run)。"  
 ::= { tempPattern 5 }

#### 4.3.2.17.3.6 临时方案剩余周期数

remainderCycleTimes OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"当前临时方案剩余周期数量。

a) 当信号机尚未执行中心临时方案，该数值应当为0；

b) 当开始执行中心临时方案，该数值应当等于临时方案运行周期数，并且在信号机执行临时方案开始后将此数值减1；

c) 当已经运行在中心临时方案状态下，信号机应当在当前每个运行周期结束，即将启动下一个运行周期时，判断此数值，如果结果大于0，则应当根据当前临时方案参数继续运行一个周期，同时将该数值减1；如果小于等于0，则应当自动退出中心控制临时方案；

d) 当已经运行在中心临时方案状态下，中心发出中心临时方案设置命令（包括修改模式、相位差、周期、方案等），信号机应当在当前周期结束后，立即同步这些参数，而不必等待前次设置的周期数运行完毕。"

::= { tempPattern 6 }

#### 4.3.2.17.3.7 临时阶段方案

tempStagePattern OBJECT IDENTIFIER

::= { tempPattern 7 }

-- 定义中心临时阶段方案命令节点

-- 此节点中包含与中心临时阶段方案控制命令相关的所有数据对象

##### 4.3.2.17.3.7.1 继承阶段方案号

inheritPatternIndex OBJECT-TYPE

SYNTAX INTEGER(1..128)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"代表中心临时方案整体参数框架继承自现有的方案的编号。

对于该对象内未包含的同时对于方案的运行是必要的属性，则直接从被继承的方案继承。

注意该参数只在周期切换时由信号机读取，期间做的任何修改不影响当前运行方案，只在下一周期发挥作用。

如果需要确定当前实际正在运行的方案，通过运行状态表的路口当前方案

((4.3.2.)13.2.3)来获取。注意当运行在中心临时方案下时，返回的模式为中心方案控制，方案号则为继承的方案。

若目前信号机实际运行的语义不是阶段语义，返回控制失败错误。"

::= { tempStagePattern 1 }

##### 4.3.2.17.3.7.2 方案协调阶段

coordinationStage OBJECT-TYPE

SYNTAX INTEGER(1..64)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"需要协调的阶段编号。中心在制定临时方案时，可以更改继承方案的协调阶段。

若目前信号机实际运行的语义不是阶段语义，返回控制失败错误。"

::= { tempStagePattern 2 }

##### 4.3.2.17.3.7.3 方案阶段时间链

split OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS                    mandatory  
DESCRIPTION

“中心临时阶段方案的阶段时间链，单位：1s。

中心在制定临时方案时，可以指定阶段的时间链。注意如果是有相位差调节需求，实际执行时信号机会根据实际情况在若干周期后实际达成指定的时间链设置，平台可通过运行状态表的协调状态信息((4.3.2.)13.2.11)来获取是否达成指定相位差。

当采用RAW格式时：

方案阶段时间链，从数组元素0开始连续两个字节依次代表方案阶段链数组中该方案的执行时间，总共32字节。

当采用JSON格式时：

方案阶段时间链，该属性是一个包含1到16个整型数字的数组，数组元素依次代表方案阶段链数组中该阶段的执行时间。

若当前信号机实际运行的语义不是阶段语义，返回控制失败错误。”

::= { tempStagePattern 3 }

#### 4.3.2.17.3.7.4 方案阶段感应链

detectChain OBJECT-TYPE

SYNTAX                    SEQUENCE OF (SEQUENCE OF INTEGER(SIZE(0..64)))  
                             SIZE(1..16)

ACCESS                    read-write

STATUS                    optional

DESCRIPTION

“平台在制定临时方案时，可以更改感应链设置。

当采用RAW格式时：

方案的阶段链中阶段的数量最大为16，Array数组中的每个元素代表一个阶段需要感应的相位，阶段需要感应的相位的长度固定为 64bit，bit0代表相位1，bit63 代表相位64，相应的 bit 位为：1 时，表示该相位为阶段感应相位。从 Array[0]到 Array[15]代表方案中阶段的执行顺序，不满16个阶段用 0 填充相应的位置。

当采用JSON格式时：

该属性是一个2维数组，第一维包含1到16个数组对象，对应方案的阶段链中的阶段，第二维包含0到64个对象，对应阶段包含相位中需要感应的相位编号。

若当前信号机实际运行的语义不是阶段语义，返回控制失败错误。”

::= { tempStagePattern 4 }

#### 4.3.2.17.3.8 临时环方案

tempRingPattern OBJECT IDENTIFIER

::={ tempPattern 8 }

-- 定义中心临时环方案命令节点

-- 此节点中包含与中心临时环方案控制命令相关的所有数据对象

##### 4.3.2.17.3.8.1 继承环方案号

inheritPatternIndex OBJECT-TYPE

SYNTAX                    INTEGER(1..128)

ACCESS                    read-write

STATUS                    optional

DESCRIPTION

“代表中心临时方案整体参数框架继承自现有的方案的编号。

对于该对象内未包含的同时对于方案的运行是必要的属性，则直接从被继承的方案继承。

注意该参数只在周期切换时由信号机读取，期间做的任何修改不影响当前运行方案，只在下一周期发挥作用。

如果需要确定当前实际正在运行的方案，通过运行状态表的路口当前环方案((4.3.2.)13.2.6)来获取。注意当运行在中心临时方案下时，返回的模式为中心方案控制，方案号则为继承的方案。

若目前信号机实际运行的语义不是环语义，返回控制失败错误。”

```
::= { tempRingPattern 1 }
```

#### 4.3.2.17.3.8.2 方案协调相位

coordinationPhase OBJECT-TYPE

SYNTAX INTEGER(1..64)

ACCESS read-write

STATUS optional

DESCRIPTION

“需要协调的相位编号。中心在制定临时方案时，可以更改继承方案的协调相位。

若目前信号机实际运行的语义不是环语义，返回控制失败错误。”

```
::= { tempRingPattern 2 }
```

#### 4.3.2.17.3.8.3 方案绿信比时间链

splitChain OBJECT-TYPE

SYNTAX SEQUENCE OF (SEQUENCE OF INTEGER(SIZE(1..16)))  
SIZE(1..8)

ACCESS read-write

STATUS optional

DESCRIPTION

“中心在制定临时方案时，可以指定相位的时间链。注意如果是有相位差调节需求，实际执行时信号机会根据实际情况在若干周期后实际达成指定的时间链设置，平台可通过运行状态表的协调状态信息((4.3.2.)13.2.11来获取是否达成指定相位差。

当采用RAW格式时：

该属性是一个8\*32字节的数组。固定8个环，对于每个环中的每个绿信比时间，用连续两个字节代表，高字节在前，低字节在后，

Array[0][0]~Array[0][31]代表方案中环1总共16个相位的绿信比时间。

当采用JSON格式时：

该属性是一个二维数组。第一维对应环数量，范围1到8，第二维是一个包含1到16个整型数据的数组，每个数据对应环中的绿信比时间。

若当前信号机实际运行的语义不是环语义，返回控制失败错误。”

```
::= { tempRingPattern 3 }
```

#### 4.3.2.17.3.8.4 方案环相位感应链

detectChain OBJECT-TYPE

SYNTAX SEQUENCE OF (SEQUENCE OF INTEGER(SIZE(1..16)))  
SIZE(1..8)

ACCESS read-write

STATUS optional

DESCRIPTION

“平台在制定临时方案时，可以更改感应链设置。

当采用RAW格式时：

环方案的相位感应链为8\*2的数组 (Byte\_Array[8][2])，环数量最大为8，每个环中相位的感应链数量最大为16。Byte\_Array[0]代表环1的相位感应链，Byte\_Array[0][0]~Byte\_Array[0][1]共16个bit对应环1中16个相位，数据结构为 BYTE2、BYTE1，当相应bit位为1时，表示该相位的检测器参与该环方案的对应相位的感应控制。

当采用JSON格式时：

该属性是一个2维数组，第一维包含1到8个数组对象，对应环方案的环数量，第二维包含0到16个对象，数量与环的相位数量一致，当该相位参与感应控制，数组元素为1，否则为0。

若当前信号机实际运行的语义不是环语义，返回控制失败错误。”

```
::= { tempRingPattern 4 }
```

#### 4.3.2.18 数据类18 命令管道

```
pipe OBJECT IDENTIFIER ::= { asc 18 }
```

-- 定义命令管道信息节点

- 此节点中包含与信号控制机命令管道信息相关的所有数据对象
- 当使用SNMP协议访问本数据类时，设备标识号路径为：  
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.  
设备.交通信号控制机.命令管道(1.3.6.1.4.1.61332.3.2.2.18)。

#### 4.3.2.18.1 命令值

```
command OBJECT-TYPE
 SYNTAX INTEGER(
 0(Order_Cancel),
 1(Order_Flash),
 2(Order_Red),
 3(Order_On),
 4(Order_Off),
 5(Order_Reset))
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "命令值定义如下：
 a) 取消命令 (Order_Cancel) 0x00
 b) 黄闪 (Order_Flash) 0x01
 c) 全红 (Order_Red) 0x02
 d) 开灯 (Order_On) 0x03
 e) 关灯 (Order_Off) 0x04
 f) 重启 (Order_Reset) 0x05"
 ::= { pipe 1 }
```

#### 4.3.2.19 数据类19 环方案信息

```
ringPattern OBJECT IDENTIFIER ::= { asc 19 }
-- 定义环方案信息节点
-- 此节点中包含与信号控制机环方案信息相关的所有数据对象
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.环方案信息(1.3.6.1.4.1.61332.3.2.2.19)。
```

##### 4.3.2.19.1 实际方案数量

```
maxPattern OBJECT-TYPE
 SYNTAX INTEGER(0..128)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "该信号机实际使用的方案编号的最大值，整数型，范围0~128。
 当使用阶段交通语义时，此项设置为0。"
 ::= { ringPattern 1 }
```

##### 4.3.2.19.2 方案配置表

```
patternCfgTable OBJECT-TYPE
 SYNTAX SEQUENCE OF patternCfgEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 "包含每个方案配置的信息数据组，数组元素数量等于maxPattern。"
 ::= { ringPattern 2 }
```

```
patternCfgEntry OBJECT-TYPE
 SYNTAX patternCfgEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
```

“每个方案的信息数据组。这是一个数据序列。”

```
INDEX { index }
::= { patternCfgTable 1 }

patternCfgEntry ::= SEQUENCE{
 index INTEGER,
 cycle INTEGER,
 coordinationIndex INTEGER,
 offset INTEGER,
 offsetSyncPoint INTEGER,
 ringNum INTEGER,
 ringCfgTable SEQUENCE OF ringCfgEntry,
 adaptiveMinCycle INTEGER,
 adaptiveMaxCycle INTEGER,
 adaptiveMaxCycleDelta INTEGER}
```

#### 4.3.2.19.2.1 方案编号

```
index OBJECT-TYPE
 SYNTAX INTEGER(1..128)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 “信号机内关于环方案的序号，整数型，方案1的编号为1，以此类推，最大128
 个方案。”
::= { patternCfgEntry 1 }
```

#### 4.3.2.19.2.2 方案周期

```
cycle OBJECT-TYPE
 SYNTAX INTEGER(1..65535)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “环方案运行周期，单位：1s。”
::= { patternCfgEntry 2 }
```

#### 4.3.2.19.2.3 方案协调序号

```
coordinationIndex OBJECT-TYPE
 SYNTAX INTEGER(1..16)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “环方案中协调相位的序号。”
::= { patternCfgEntry 3 }
```

#### 4.3.2.19.2.4 方案相位差时间

```
offset OBJECT-TYPE
 SYNTAX INTEGER(0..65535)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 “相位差时间，单位：1s。”
::= { patternCfgEntry 4 }
```

#### 4.3.2.19.2.5 方案相位差计算参考点

```
offsetSyncPoint OBJECT-TYPE
 SYNTAX INTEGER(
 0(Offset_Ref_Unavailable),
 1(Offset_Ref_GreenOn),
 2(Offset_Ref_GreenOff),
 3(Offset_Ref_YellowOff))
 ACCESS read-write
 STATUS optional
```

## DESCRIPTION

“相位差计算参考点类型定义如下：

- |                                      |      |
|--------------------------------------|------|
| a) 未设置 (Offset_Ref_Unavailable)      | 0x00 |
| b) 相对协调相位绿灯开始 (Offset_Ref_GreenOn)   | 0x01 |
| c) 相对协调相位绿灯结束 (Offset_Ref_GreenOff)  | 0x02 |
| d) 相对协调相位黄灯结束 (Offset_Ref_YellowOff) | 0x03 |

**注：**本标准参考GB/T 20999-2017，对于相位过渡的定义为“关闭、启动路权过渡灯色1、2、3”，逻辑严密的角度，应当将此类型定义为相对“关闭、启动路权过渡灯色1、2、3”，但考虑到此种方式难于理解并且不符合一般习惯，因此仍然定义为符合日常习惯的相对灯色的定义。”

```
::= { patternCfgEntry 5 }
```

## 4.3.2.19.2.6 方案环数量

ringNum OBJECT-TYPE

SYNTAX INTEGER(1..8)  
ACCESS read-write  
STATUS optional  
DESCRIPTION

“该环方案的环数量，常见为2，代表双环，最大为8。”

```
::= { patternCfgEntry 6 }
```

## 4.3.2.19.2.7 方案环配置表

ringCfgTable OBJECT-TYPE

SYNTAX SEQUENCE OF ringCfgEntry  
ACCESS not-accessible  
STATUS optional  
DESCRIPTION

“包含每个环配置的信息数据组，数组元素数量等于ringNum。”

```
::= { patternCfgTable 7 }
```

ringCfgEntry OBJECT-TYPE

SYNTAX ringCfgEntry  
ACCESS not-accessible  
STATUS optional  
DESCRIPTION

“每个环的信息数据组。这是一个数据序列。”

INDEX { index }

```
::= { ringCfgTable 1 }
```

```
ringCfgEntry ::= SEQUENCE{
```

|                |                      |
|----------------|----------------------|
| index          | INTEGER,             |
| phaseChain     | SEQUENCE OF INTEGER, |
| phaseTypeChain | SEQUENCE OF INTEGER, |
| splitChain     | SEQUENCE OF INTEGER, |
| detectorChain  | SEQUENCE OF INTEGER) |

## 4.3.2.19.2.7.1 环的编号

index OBJECT-TYPE

SYNTAX INTEGER(1..8)  
ACCESS read-only  
STATUS optional  
DESCRIPTION

“环的编号，整数型，环1的编号为1，以此类推，最大等于ringNum。”

```
::= { ringCfgEntry 1 }
```

## 4.3.2.19.2.7.2 环的相位链

phaseChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))  
ACCESS read-write  
STATUS optional

## DESCRIPTION

“当采用RAW格式定义时：

该属性是含16个字节的数组，单个环的相位链中相位的数量最大为16，数组中的每个元素代表一个相位编号，从Array[0]到Array[15]代表方案中环相位的执行顺序，不满16个相位用0填充相应的位。

当采用JSON格式定义时：

该属性是含1到16个整型数据的数组，每个数据对应相位的编号。”

```
::= { ringCfgEntry 2 }
```

## 4.3.2.19.2.7.3 环的相位出现类型链

phaseTypeChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS optional

## DESCRIPTION

“当采用RAW格式定义时：

该属性是含16个字节的数组，单个环中相位出现类型，数量最大为16，数组中的每个元素代表一个相位出现类型，从Array[0]到Array[15]代表方案中环1各个相位的相位出现类型。环中不满16个相位的用0填充。

当采用JSON格式定义时：

该属性是含1到16个整型数据的数组，每个数据对应环中每个相位的出现类型。

相位出现类型定义如下：

a) 固定出现 (Type\_Fix) 0x10

b) 按需求出现 (Type\_Demand) 0x20

**注：**相位出现类型主要规定环方案中相位出现类型：固定出现是指在每个信号周期内都会出现；按需求出现是指在每个信号周期内根据其检测器上的需求确定是否出现。”

```
::= { ringCfgEntry 3 }
```

## 4.3.2.19.2.7.4 环的绿信比时间链

splitChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS optional

## DESCRIPTION

“当采用RAW格式时：

该属性是一个32字节的数组，环中的每个绿信比时间，用连续两个字节代表，高字节在前，低字节在后，Byte\_Array[0]~Byte\_Array[1]代表环的第1个相位绿信比时间，Byte\_Array[2]~Byte\_Array[3]代表环的第2个相位绿信比时间，以此类推。环中不满16个相位的用0填充。

当采用JSON格式时：

该属性是含1到16个整型数字的数组，对应环中每个相位绿信比时间。”

```
::= { ringCfgEntry 4 }
```

## 4.3.2.19.2.7.5 环的相位感应链

detectorChain OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..16))

ACCESS read-write

STATUS optional

## DESCRIPTION

“当采用RAW格式时：

该属性是一个2字节的数组，共16bit，每个bit对应环中16个相位，数据结构为 BYTE2、BYTE1，当相应bit位为1时，表示该相位的检测器参与该环方案的对应相位的感应控制。

当采用JSON格式时：

该属性是一个16个BIT的数组，每个bit对应环中16个相位，当相应bit位为1时，表示该相位的检测器参与该环方案的对应相位的感应控制。”

```
::= { ringCfgEntry 5 }
```

#### 4.3.2.19.2.8 自适应最小周期

```
adaptiveMinCycle OBJECT-TYPE
 SYNTAX INTEGER(1..65535)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "自适应控制最小周期时间，单位：1s。"
 ::= { patternCfgEntry 8 }
```

#### 4.3.2.19.2.9 自适应最大周期

```
adaptiveMaxCycle OBJECT-TYPE
 SYNTAX INTEGER(1..65535)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "自适应控制最大周期时间，单位：1s。"
 ::= { patternCfgEntry 9 }
```

#### 4.3.2.19.2.10 自适应最大周期波动

```
adaptiveMaxCycleDelta OBJECT-TYPE
 SYNTAX INTEGER(1..65535)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "自适应控制最大周期波动时间，单位：1s。"
 ::= { patternCfgEntry 10 }
```

#### 4.3.2.20 数据类20 跟随相位信息

```
overlap OBJECT IDENTIFIER ::= { asc 20 }
-- 定义跟随相位信息节点
-- 此节点中包含与信号控制机跟随相位信息相关的所有数据对象
-- 当使用SNMP协议访问本数据类时，设备标识号路径为：
iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.
设备.交通信号控制机.跟随相位信息(1.3.6.1.4.1.61332.3.2.2.20)。
```

##### 4.3.2.20.1 实际配置跟随相位数量

```
maxOverlap OBJECT-TYPE
 SYNTAX INTEGER(0..64)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "信号机中已经使用的最大的跟随相位编号。"
 ::= { overlap 1 }
```

##### 4.3.2.20.2 跟随相位配置表

```
overlapCfgTable OBJECT-TYPE
 SYNTAX SEQUENCE OF overlapCfgEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 "包含每个跟随相位配置的信息数据组，数组元素数量等于maxOverlap。"
 ::= { overlap 2 }
```

```
overlapCfgEntry OBJECT-TYPE
 SYNTAX overlapCfgEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
```

“每个跟随相位的信息数据组。这是一个数据序列。”

```
INDEX { index }
::= { overlapCfgTable 1 }

overlapCfgEntry ::= SEQUENCE{
 index INTEGER,
 mode INTEGER,
 channelIncluded SEQUENCE OF INTEGER,
 includedPhase SEQUENCE OF INTEGER,
 modifierPhase SEQUENCE OF INTEGER,
 lateBeginTime INTEGER,
 earlyEndTime INTEGER,
 trailingTime INTEGER}
```

#### 4.3.2.20.2.1 跟随相位编号

```
index OBJECT-TYPE
 SYNTAX INTEGER(1..64)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
```

“信号机中跟随相位的序号，跟随相位1的跟随相位编号固定为1，以此类推。”

```
::= { overlapCfgEntry 1 }
```

#### 4.3.2.20.2.2 跟随模式

```
mode OBJECT-TYPE
 SYNTAX INTEGER(
 0(Overlap_Type_Unavailable),
 1(Overlap_Type_Normal),
 2(Overlap_Type_MinusGreenYellow),
 255(Overlap_Type_Undefined))
 ACCESS read-write
 STATUS optional
 DESCRIPTION
```

“跟随模式设置，定义如下：

a) 未设置 (Overlap\_Type\_Unavailable)  
0x00

b) 正常跟随模式 (Overlap\_Type\_Normal)  
0x01

跟随相位的输出受跟随的相位编号参数控制。

- 1) 有下列情形时跟随相位输出绿灯：
  - 当跟随的母相位编号包含的相位至少有一个是绿灯时；
  - 当跟随的母相位编号包含的相位至少有一个是黄灯（或者全红清空）且跟随的母相位编号包含下一相位时；
- 2) 如果跟随的母相位编号包含的相位至少有一个是黄灯且跟随的母相位编号不包含下一相位，跟随相位输出黄灯。
- 3) 如果跟随的母相位编号包含的所有相位的绿灯和黄灯无效，将输出红灯。

c) 修正跟随模式 (Overlap\_Type\_MinusGreenYellow)  
0x02

此种操作类型时，跟随相位的输出受跟随的母相位编号和跟随的修正相位编号参数控制。

- 1) 有下列情形时跟随相位输出绿灯：
  - 当跟随的母相位编号包含的相位至少有一个是绿灯且跟随的修正相位编号包含的相位全部不是绿灯时；
  - 当跟随的母相位编号包含的相位至少有一个是黄灯（或者全红清空）且跟随的母相位编号包含下一相位且跟随的修正相位编号包含的相位全部不是绿灯时；

- 2) 如果跟随的母相位编号包含的相位至少有一个是黄灯且跟随的修正相位编号包含的相位全部不是黄灯且跟随的母相位编号不包含下一相位, 跟随相位输出黄灯。
- 3) 如果跟随的母相位编号包含的所有相位的绿灯和黄灯无效, 将输出红灯。

d) 未在此标准定义的跟随模式 (Overlap\_Type\_Undefined)

0xFF"

::= { overlapCfgEntry 2 }

#### 4.3.2.20.2.3 跟随相位的通道

channelIncluded OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..64))

ACCESS read-write

STATUS optional

DESCRIPTION

"表示该跟随相位对应的通道。

当采用RAW格式时, 跟随相位的通道的长度为8个字节(64 bit), bit0代表信号通道1, bit63代表信号通道64, 数据结构为BYTE8、...、BYTE1, 代表 bit63~bit56、...、bit7~bit0, 相应的 bit 位为1 时, 表示该跟随相位会从该通道上输出。

当采用JSON格式时, 跟随相位的通道是一个包含0~64个整型数据的数组, 数组中每个数据代表该跟随相位包含的一个通道的编号, 当跟随相位不包含实际的通道时, 该数组长度为0。"

::= { overlapCfgEntry 3 }

#### 4.3.2.20.2.4 跟随的母相位编号

includedPhase OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(1..64))

ACCESS read-write

STATUS optional

DESCRIPTION

"表示该跟随相位跟随的相位。

当采用RAW格式时, 跟随的母相位编号长度固定为8个字节(64 bit), bit0代表相位1, bit63代表相位64, 数据结构为 BYTE8、...BYTE1, 代表 bit63~bit56、...bit7~bit0, 相应的 bit 位为1时, 表示该相位包含在跟随对象中。

当采用JSON格式时, 跟随的母相位编号为包含1到64个整型数据的数组, 每个数据对应跟随的母相位的编号。"

::= { overlapCfgEntry 4 }

#### 4.3.2.20.2.5 跟随的修正相位编号

modifierPhase OBJECT-TYPE

SYNTAX SEQUENCE OF INTEGER(SIZE(0..64))

ACCESS read-write

STATUS optional

DESCRIPTION

"表示跟随相位的修正相位。

当采用RAW格式时, 跟随的修正相位编号长度固定为8个字节(64 bit), bit0代表相位1, bit63代表相位64, 数据结构为 BYTE8、...BYTE1, 代表 bit63~bit56、...bit7~bit0, 相应的 bit 位为1时, 表示该相位包含在跟随修正对象中。

当采用JSON格式时, 跟随的修正相位编号为包含0到64个整型数据的数组, 每个数据对应跟随的修正相位的编号。"

::= { overlapCfgEntry 5 }

#### 4.3.2.20.2.6 跟随相位的晚启动时间

lateBeginTime OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS optional

## DESCRIPTION

“跟随相位相对于被跟随相位开始时刻晚起动的秒数。”

::= { overlapCfgEntry 6 }

## 4.3.2.20.2.7 跟随相位的早结束时间

earlyEndTime OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS optional

## DESCRIPTION

“跟随相位相对于被跟随相位开始时刻早结束的秒数。”

::= { overlapCfgEntry 7 }

## 4.3.2.20.2.8 跟随相位路权延长时间

trailingTime OBJECT-TYPE

SYNTAX INTEGER(0..255)

ACCESS read-write

STATUS optional

## DESCRIPTION

“0-255秒，如果此参数大于0且overlap的绿灯正常结束，跟随相位路权将延长此参数设定的秒数。”

::= { overlapCfgEntry 7 }

## 4.3.2.21 数据类21 通道时段控制计划信息

channelTimePlan OBJECT IDENTIFIER ::= { asc 21 }

-- 定义通道时段控制计划信息节点

-- 此节点中包含与信号控制机通道时段控制计划信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.  
设备.交通信号控制机.通道时段控制计划信息(1.3.6.1.4.1.61332.3.2.2.21)。

## 4.3.2.21.1 实际通道时段控制计划表数量

maxPlan OBJECT-TYPE

SYNTAX INTEGER(0..16)

ACCESS read-only

STATUS optional

## DESCRIPTION

“信号机中已经使用的最大的通道时段控制计划表编号。”

::= { channelTimeCtl 1 }

## 4.3.2.21.2 通道时段控制计划表

planCfgTable OBJECT-TYPE

SYNTAX SEQUENCE OF planCfgEntry

ACCESS not-accessible

STATUS optional

## DESCRIPTION

“包含每个通道时段控制配置的信息数据组，数组元素数量等于maxPlan。”

::= { channelTimeCtl 2 }

planCfgEntry OBJECT-TYPE

SYNTAX planCfgEntry

ACCESS not-accessible

STATUS optional

## DESCRIPTION

“每个通道时段控制的信息数据组。这是一个数据序列。”

INDEX { index }

::= { planCfgTable 1 }

planCfgEntry ::= SEQUENCE{

index INTEGER,

```

maxChannel INTEGER,
timeSlotTable SEQUENCE OF timeSlotEntry)

```

#### 4.3.2.21.2.1 计划表编号

```

index OBJECT-TYPE
 SYNTAX INTEGER(1..16)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 "通道时段控制计划表的序号。"
 ::= { planCfgEntry 1 }

```

#### 4.3.2.21.2.2 通道数量

```

maxChannel OBJECT-TYPE
 SYNTAX INTEGER(0..64)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "表示参与该计划的通道的最大编号。"
 ::= { planCfgEntry 2 }

```

#### 4.3.2.21.2.3 通道时段配置表

```

timeSlotTable OBJECT-TYPE
 SYNTAX SEQUENCE OF timeSlotEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 "包含每个通道时段配置的信息数据组，数组元素数量等于maxChannel。"
 ::= { planCfgEntry 3 }

```

```

planCfgEntry OBJECT-TYPE
 SYNTAX planCfgEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 "每个通道时段配置的信息数据组。这是一个数据序列。"
 INDEX { index }
 ::= { timeSlotTable 1 }

```

```

timeSlotEntry ::= SEQUENCE{
 index INTEGER,
 enable INTEGER,
 maxTimeSlot INTEGER,
 beginTimeChain SEQUENCE OF INTEGER,
 endTimeChain SEQUENCE OF INTEGER,
 controlChain SEQUENCE OF INTEGER)

```

#### 4.3.2.21.2.3.1 通道编号

```

index OBJECT-TYPE
 SYNTAX INTEGER(1..64)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 "通道的序号，最大64个通道。"
 ::= { timeSlotEntry 1 }

```

#### 4.3.2.21.2.3.2 配置标记

```

enable OBJECT-TYPE
 SYNTAX INTEGER(0,1)
 ACCESS read-write
 STATUS optional
 DESCRIPTION

```

“该时段配置表配置使能标记。0：不使能；1：使能”  
 ::= { timeSlotEntry 2 }

#### 4.3.2.21.2.3.3 时段数量

maxTimeSlot OBJECT-TYPE  
 SYNTAX INTEGER(0..48)  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 “该时段配置表实际包含的时段数量，最大48个时段。”  
 ::= { timeSlotEntry 3 }

#### 4.3.2.21.2.3.4 时段开始时间链

beginTimeChain OBJECT-TYPE  
 SYNTAX SEQUENCE OF INTEGER(SIZE(0..96))  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 “时段开始详细设置，协议约定一天最大48个时段，每2个字节表示一个时间，其中Byte Array[1]表示时钟，Byte Array[0]表示分钟。  
 当采用RAW格式时，为固定96个字节的数组；  
 当采用JSON格式时，为包含多个字节的数组，每两个字节对应一个时段，时段总数等于maxTimeSlot。”  
 ::= { timeSlotEntry 4 }

#### 4.3.2.21.2.3.5 时段结束时间链

endTimeChain OBJECT-TYPE  
 SYNTAX SEQUENCE OF INTEGER(SIZE(0..96))  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 “时段结束详细设置，协议约定一天最大48个时段，每2个字节表示一个时间，其中Byte Array[1]表示时钟，Byte Array[0]表示分钟。  
 当采用RAW格式时，为固定96个字节的数组；  
 当采用JSON格式时，为包含多个字节的数组，每两个字节对应一个时段，时段总数等于maxTimeSlot。”  
 ::= { timeSlotEntry 5 }

#### 4.3.2.21.2.3.6 时段控制链

controlChain OBJECT-TYPE  
 SYNTAX SEQUENCE OF INTEGER(SIZE(0..48))  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 “时段控制数组，元素数量等于maxTimeSlot，每个元素为通道控制状态，取值参见附录A.3灯组状态。  
 当采用RAW格式时：  
 该属性是一个48字节数组。每个字节对应一个时段控制，不使用的字节填0。  
 当采用JSON格式时：  
 该属性是一个包含1到48个整型数据的数组，成员数量等于maxTimeSlot，每个成员对应一个时段的控制。”  
 ::= { timeSlotEntry 6 }

#### 4.3.2.22 数据类22 通道时段控制调度信息

channelTimeCtlShedule OBJECT IDENTIFIER ::= { asc 22 }  
 -- 定义通道时段控制调度表信息节点  
 -- 此节点中包含与信号控制机通道时段控制调度表信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.  
设备.交通信号控制机.通道时段控制调度信息(1.3.6.1.4.1.61332.3.2.2.22)。

#### 4.3.2.22.1 实际通道时段控制调度表数量

```
maxShedule OBJECT-TYPE
 SYNTAX INTEGER(0..16)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 "信号机中已经使用的最大的通道时段调度计划表编号。"
 ::= { channelTimeCtlShedule 1 }
```

#### 4.3.2.22.2 通道时段控制调度表

```
sheduleCfgTable OBJECT-TYPE
 SYNTAX SEQUENCE OF sheduleCfgEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 "包含每个通道时段调度配置的信息数据组，数组元素数量等于maxShedule。"
 ::= { channelTimeCtlShedule 2 }
```

```
sheduleCfgEntry OBJECT-TYPE
 SYNTAX sheduleCfgEntry
 ACCESS not-accessible
 STATUS optional
 DESCRIPTION
 "每个通道时段调度的信息数据组。这是一个数据序列。"
 INDEX { index }
 ::= { sheduleCfgTable 1 }
```

```
sheduleCfgEntry ::= SEQUENCE{
 index INTEGER,
 priority INTEGER,
 day INTEGER,
 month INTEGER,
 date INTEGER,
 planIndex INTEGER}
```

#### 4.3.2.22.2.1 调度表编号

```
index OBJECT-TYPE
 SYNTAX INTEGER(1..16)
 ACCESS read-only
 STATUS optional
 DESCRIPTION
 "通道时段控制调度表的序号。"
 ::= { sheduleCfgEntry 1 }
```

#### 4.3.2.22.2.2 调度优先级

```
priority OBJECT-TYPE
 SYNTAX INTEGER(0..255)
 ACCESS read-write
 STATUS optional
 DESCRIPTION
 "通道控制调度表的优先级，值越小优先级越高，0的优先级最高。"
 ::= { sheduleCfgEntry 2 }
```

#### 4.3.2.22.2.3 调度表星期值

```
day OBJECT-TYPE
 SYNTAX INTEGER(0..255)
 ACCESS read-write
```

STATUS optional  
DESCRIPTION

“bit0表示星期天，当bit0为1时，表示该调度表星期天需要执行，bit1为星期一，以此类推，bit7保留。”

::= { sheduleCfgEntry 3 }

#### 4.3.2.22.2.4 调度表月份值

month OBJECT-TYPE

SYNTAX INTEGER(0..65535)

ACCESS read-write

STATUS optional

DESCRIPTION

“16位整型数。Bit0保留，bit1为1时，表示该调度表1月需要执行，以此类推，bit13~bit15保留。”

::= { sheduleCfgEntry 4 }

#### 4.3.2.22.2.5 调度表日期值

date OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS optional

DESCRIPTION

“32位整型数。当bit0为1时，表示该调度表1日需要执行，以此类推，bit30表示31日，bit31保留。”

::= { sheduleCfgEntry 5 }

#### 4.3.2.22.2.6 通道时段控制计划

planIndex OBJECT-TYPE

SYNTAX INTEGER(1..16)

ACCESS read-write

STATUS optional

DESCRIPTION

“该调度表执行的通道时段控制计划表的编号。”

::= { sheduleCfgEntry 5 }

#### 4.3.2.23 数据类23 用户信息

user OBJECT IDENTIFIER ::= { asc 23 }

-- 定义用户信息节点

-- 此节点中包含与信号控制机用户信息相关的所有数据对象

-- 当使用SNMP协议访问本数据类时，设备标识号路径为：

iso.org.dod.internet.private.enterprise.道路交通安全研究中心.交通.设备.交通信号控制机.用户信息(1.3.6.1.4.1.61332.3.2.2.23)。

##### 4.3.2.23.1 实际用户数

maxUser OBJECT-TYPE

SYNTAX INTEGER(0..64)

ACCESS read-only

STATUS optional

DESCRIPTION

“信号机中已经使用的最大的用户编号。”

::= { user 1 }

##### 4.3.2.23.2 用户信息表

userInfoTable OBJECT-TYPE

SYNTAX SEQUENCE OF userInfoEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

“包含每个用户配置的信息数据组，数组元素数量等于maxUser。”

```
::= { user 2 }
```

```
userInfoEntry OBJECT-TYPE
```

```
SYNTAX userInfoEntry
ACCESS not-accessible
STATUS optional
DESCRIPTION
```

“每个用户配置的信息数据组。这是一个数据序列。”

```
INDEX {index}
```

```
::= { userInfoTable 1 }
```

```
userInfoEntry ::= SEQUENCE{
```

```
index INTEGER,
name OCTET STRING,
key OCTET STRING,
authority INTEGER,
lastAccessTime INTEGER)
```

#### 4.3.2.23.2.1 用户编号

```
index OBJECT-TYPE
```

```
SYNTAX INTEGER(1..64)
ACCESS read-only
STATUS optional
DESCRIPTION
```

“信号机中用户的序号，用户1的用户编号固定为1，以此类推。”

```
::= { userInfoEntry 1 }
```

#### 4.3.2.23.2.2 用户名称

```
name OBJECT-TYPE
```

```
SYNTAX OCTET STRING(SIZE(8..16))
ACCESS read-write
STATUS optional
DESCRIPTION
```

“用户的名称，该名称对应消息帧中的用户名字段

当采用RAW格式时，该属性是一个16字节的数组，不满16字节的以0填充；

当采用JSON格式时，该属性是一个字符串，长度范围为8到16。”

```
::= { userInfoEntry 2 }
```

#### 4.3.2.23.2.3 用户密钥

```
key OBJECT-TYPE
```

```
SYNTAX OCTET STRING(SIZE(8..16))
ACCESS read-write
STATUS optional
DESCRIPTION
```

“用户的密钥，用于用户身份验证。

当采用RAW格式时，该属性是一个16字节的数组，不满16字节的以0填充；

当采用JSON格式时，该属性是一个字符串，长度范围为8到16。”

```
::= { userInfoEntry 3 }
```

#### 4.3.2.23.2.4 用户权限

```
authority OBJECT-TYPE
```

```
SYNTAX INTEGER(0..255)
ACCESS read-write
STATUS optional
DESCRIPTION
```

“用户的权限，详见附录A.2用户权限。”

```
::= { userInfoEntry 4 }
```

#### 4.3.2.23.2.5 最近访问时间

```
accessTime OBJECT-TYPE
```

```
SYNTAX Counter64(0..9999999999999999)
```

ACCESS read-only  
STATUS optional  
DESCRIPTION

“用户的最近一次访问的时间。

相对1970年1月1日0时0分0毫秒的毫秒计时，13位的十进制数。

当采用RAW格式时，时间为8字节（Int64），高字节在前，低字节在后。

当采用JSON格式时，时间为13位的整形数据。”

::= { userInfoEntry 5 }

## 附录 A

### (规范性附录)

### 通信协议规范定义

#### A.1 概述

通信协议规范规定的数值定义可扩展，各厂家可以根据需求定义。

对象名称后括号内的英文名称可用于生成JSON格式数据字段内容。

#### A.2 用户权限

a) 协议约定针对不同类型的用户进行不同等级的授权，用户只有在授权范围内可以访问数据并进行相应的控制和设置，不在访问授权范围内的，将返回无访问权限错误代码；

b) 协议约定用户应当设置密钥，如果密钥和系统保存密钥不符，将返回密钥错误代码；

c) 协议约定信号机回复或者主动推送的报文使用的用户名与查询或者设置报文的用户名保持一致，并且应当按照配置的加密方式采用对应的加密报文回复；

用户权限定义为一个8位无符号数，其中每个Bit对应一种用户权限，同一个用户可以获得多种用户权限，此时只要将权限对应Bit位置为1，无权限的对应Bit位置为0，各Bit位对应权限定义如下：

##### Bit 0 - 设备管理权限 (Permission\_Device\_Manage)

该权限可实现数据类：设备信息、基础信息、通道信息、相位信息、检测器信息、命令管道、检测器设备信息、相位安全信息的读取和设置。

该权限可实现数据类：运行状态信息、交通数据信息、报警数据信息、故障数据信息的读取。

##### Bit 1 - 方案管理权限 (Permission\_Plan\_Manage)

该权限可实现数据类：阶段信息、紧急优先信息、阶段方案信息、过渡约束信息、日计划信息、调度信息、命令管道、环方案信息、跟随相位信息、通道时段控制计划信息、通道时段控制调度信息的读取和设置。

该权限可实现数据类：设备信息、基础信息、通道信息、相位信息、检测器信息、检测器设备信息、相位安全信息的读取。

该权限可实现数据类：运行状态信息、交通数据信息、报警数据信息、故障数据信息的读取。

##### Bit 2 - 中心访问权限 (Permission\_Center\_Read)

该权限可实现数据类：中心控制、命令管道的读取。

该权限可实现数据类：阶段信息、紧急优先信息、阶段方案信息、过渡约束信息、日计划信息、调度信息、环方案信息、跟随相位信息、通道时段控制计划信息、通道时段控制调度信息的读取。

该权限可实现数据类：设备信息、基础信息、通道信息、相位信息、检测器信息、检测器设备信息、相位安全信息的读取。

该权限可实现数据类：运行状态信息、交通数据信息、报警数据信息、故障数据信息的读取。

##### Bit 3 - 中心控制权限 (Permission\_Center\_Control)

该权限可实现数据类：中心控制、命令管道的读取和设置。

该权限可实现数据类：阶段信息、紧急优先信息、阶段方案信息、过渡约束信息、日计划信息、调度信息、环方案信息、跟随相位信息、通道时段控制计划信息、通道时段控制调度信息的读取。

该权限可实现数据类：设备信息、基础信息、通道信息、相位信息、检测器信息、检测器设备信息、相位安全信息的读取。

该权限可实现数据类：运行状态信息、交通数据信息、报警数据信息、故障数据信息的读取。

#### Bit 4 - 边缘只读权限 (Permission\_Edge\_Read)

该权限可实现数据类：阶段信息、阶段方案信息、环方案信息、跟随相位信息的读取。

该权限可实现数据类：设备信息、基础信息、通道信息、相位信息的读取。

该权限可实现数据类：运行状态信息、报警数据信息、故障数据信息的读取。

#### Bit 5 ~ Bit 6 - 保留

#### Bit 7 - 用户管理权限 (Permission\_User\_Manage)

该权限可实现数据类：用户信息的读取和更改。

**注1：**当该数据置为0xff时代表获取所有控制权限。

**注2：**对于本协议未涵盖的数据类，可根据实际需求定义相应权限。

**注3：**部分数据类内的个别属性对于访问权限有特殊安排的，可以在附录B（交通信号控制对象标识号定义）中详细描述。

### A.3 灯组状态

灯组状态定义如下：

- a) 未设置 (Light\_Status\_Unavailable)  
0x00
- b) 灭灯 (Light\_Status\_Off)  
0x01
- c) 红灯 (Light\_Status\_Red)  
0x10
- d) 红闪 (Light\_Status\_RedFlash)  
0x11
- e) 红快闪 (Light\_Status\_RedFastFlash)  
0x12
- f) 绿灯 (Light\_Status\_Green)  
0x20
- g) 绿闪 (Light\_Status\_GreenFlash)  
0x21
- h) 绿快闪 (Light\_Status\_GreenFastFlash)  
0x22
- i) 黄灯 (Light\_Status\_Yellow)  
0x30
- j) 黄闪 (Light\_Status\_YellowFlash)  
0x31
- k) 黄快闪 (Light\_Status\_YellowFastFlash)  
0x32
- l) 红黄灯 (Light\_Status\_RedYellow)  
0x40

- m) 本标准未涵盖的类型 (Light\_Status\_Undefined)  
0xFF

#### A.4 信号机运行模式

信号机运行模式定义如下：

- a) 未设置  
0x00
- b) 中心控制模式 (Mode\_Center\_Control)  
0x10
- |                                                                                         |      |
|-----------------------------------------------------------------------------------------|------|
| 1) 中心日计划控制 (Mode_Center_TimeTable_Control)                                              | 0x11 |
| 2) 中心优化控制 (Mode_Center_Optimization_Control)                                            |      |
| 0x12                                                                                    |      |
| 3) 中心协调控制 (Mode_Center_Coordination_Control)                                            |      |
| 0x13                                                                                    |      |
| 4) 中心自适应控制 (Mode_Center_Adaptive_Control)                                               | 0x14 |
| 5) 中心锁定控制 (Mode_Center_Lock_Control)                                                    |      |
| 0x15                                                                                    |      |
| 注：GB/T 20999-2017中描述为中心手动控制(Mode_Center_Manual_Control)，该定义比较宽泛，存在歧义，不利于实际执行，因此以锁定控制替代。 |      |
| 6) 中心黄闪控制 (Mode_Center_Flash_Control)                                                   | 0x16 |
| 7) 中心全红控制 (Mode_Center_AllRed_Control)                                                  |      |
| 0x17                                                                                    |      |
| 8) 中心关灯控制 (Mode_Center_AllOff_Control)                                                  |      |
| 0x18                                                                                    |      |
| 9) 中心感应控制 (Mode_Center_Va_Control)                                                      | 0x19 |
| 10) 中心方案控制 (Mode_Center_Plan_Control)                                                   |      |
| 0x1A                                                                                    |      |
| 11) 中心手动控制—灯组锁定 (Mode_Center_LightGroups_Control)                                       |      |
| 0x1B                                                                                    |      |
| 12) 中心手动控制—自动锁定 (Mode_Center_HoldLock_Control)                                          |      |
| 0x1C                                                                                    |      |
| 13) 中心手动控制—手动清空锁定 (Mode_Center_Clear_Control)                                           |      |
| 0x1D                                                                                    |      |
| 14) 中心手动控制—手动瓶颈锁定 (Mode_Center_BottleNeck_Control)                                      |      |
| 0x1E                                                                                    |      |
- c) 本地控制模式 (Mode\_Local\_Control)  
0x20
- |                                                |      |
|------------------------------------------------|------|
| 1) 本地定周期控制 (Mode_Local_FixCycle_Control)       | 0x21 |
| 2) 本地感应控制 (Mode_Local_Va_Control)              |      |
| 0x22                                           |      |
| 3) 本地协调控制 (Mode_Local_Coordination_Control)    | 0x23 |
| 4) 本地自适应控制 (Mode_Local_Adaptive_Control)       | 0x24 |
| 5) 本地手动控制 (Mode_Local_Manual_Control)          |      |
| 0x25                                           |      |
| 6) 本地手动黄闪控制 (Mode_Local_Manual_Flash_Control)  | 0x26 |
| 7) 本地手动全红控制 (Mode_Local_Manual_AllRed_Control) |      |
| 0x27                                           |      |
| 8) (公交)车辆优先控制 (Mode_Vehicle_Priority_Control)  |      |
| 0x28                                           |      |

- d) 特殊控制 (Mode\_Special\_Control)
- |                                       |      |
|---------------------------------------|------|
| 0x30                                  |      |
| 1) 黄闪控制 (Mode_Special_Flash_Control)  | 0x31 |
| 2) 全红控制 (Mode_Special_AllRed_Control) |      |
| 0x32                                  |      |
| 3) 关灯控制 (Mode_Special_AllOff_Control) |      |
| 0x33                                  |      |

注：运行模式分为3大类：中心控制模式、本地控制模式、特殊控制模式。每一个大类的控制模式下可以根据信号机内部的运行规则分成不同的小类运行模式。

## 附录 B

### (规范性附录)

### 通信规程及报文示例

#### B.1 通信规程

参见总则附录E。

信号机与上位机采用总则定义的协议或者SNMP v3协议实现通信交互。

#### B.2 用户访问授权

- a) 当上位机身份验证的用户名不在信号机内部保存的用户清单中，信号机返回无效用户 错误(Status\_InvalidUser)；
- b) 当上位机身份验证的密钥和信号机内部保存的对应用户的密钥不匹配，信号机返回密钥错误 (Status\_InvalidKey)；
- c) 当已经通过身份验证的上位机用户访问操作超出授权范围，信号机返回无访问权限错误(Status\_AccessDenied)；
- d) 当采用SNMP v3协议，上位机与信号机的访问授权模式参考SNMP v3规则实现。

#### B.3 单台信号机控制多路口流程

- a) 上位机访问单台信号机控制的多个路口时，不同路口上位机通过不同的协议端口号区分，端口号的配置通过对象标识2.3.3实现；
- b) 上位机访问第一个路口使用的端口号也是上位机访问信号机的端口号，即如果修改了上位机访问第一个路口的端口号，则数据类1通用数据中的支持的协议标准端口列表中相应的数据读取将同步变化；
- c) 上位机采用不同的协议访问对象标识2.3.3时，实际交互对象为该协议下的访问端口，即信号机内部应当独立存储多种协议下的访问端口配置，但是通过对对象标识读取或者修改时，实际读取或者修改对象为当时访问协议对应的端口配置；
- d) 当单台信号机控制多个路口时，多个路口相位、方案、计划等是独立处理，即每个路口都具有自己独立的编号方案，不同路口之间编号可以重复；
- e) 通道编号是全局唯一的，不可重复，相位和通道绑定时，如果出现不同路口绑定了同一个通道，信号机将报错误；
- f) 检测器编号是全局唯一的，检测器和相位绑定或者和车道号绑定时，如果出现不同路口绑定了同一个检测器，信号机将报错误；
- g) 同一台信号机控制多个路口时，每个路口使用的交通语义应当保持一致；
- h) 对于全局性的配置参数，例如时间（后面再补充），只能够通过第一个路口访问来修改，其他路口的访问为只读属性。

#### B.4 总则定义数据帧报文示例

一般报文格式参见总则相关章节。

本章节描述特殊类型的报文操作。

为简化，报文中版本位、安全参数、身份认证标识、用户名、认证码、协议标识字段未列出。

#### B.4.1 设置类型 – 清空

示例：

现要清空检测器11的配置，设置报文见表B.1。

表B.1 设置清空报文

| 字段名   | 值            | 长度(字节) | 描述                     |
|-------|--------------|--------|------------------------|
| 上位机ID | 0x12345678   | 4      | 视实际情况                  |
| 设备ID  | 0x87654321   | 4      | 视实际情况                  |
| 数据帧标识 | xxxx         | 2      | 视实际情况                  |
| 时间戳   | xxxxxxxxxxxx | 6      | 视实际情况                  |
| 帧类型   | 0x20         | 1      | 表示设置数据                 |
| 数据值数量 | 0x0001       | 2      | 该条报文操作1个数据值            |
| 数据值索引 | 0x0001       | 2      | 第1个数据值序号               |
| 标识号长度 | 0x04         | 1      | 该对象标识号长度为4             |
| 一级标识号 | 0x05         | 1      | 一级对象标识号5，对应：检测器信息      |
| 二级标识号 | 0x02         | 1      | 二级对象标识号2，对应：检测器配置表     |
| 三级标识号 | 0x00         | 1      | 三级对象标识号0，对应：检测器三级所有标识号 |
| 四级标识号 | 0x0B         | 1      | 通道11                   |
| 数据值长度 | 0x0001       | 2      | 第1个数据值的长度              |
| 数据值   | 0x00         | 1      | 值为0，代表清空操作             |

清空成功后，再次查询检测器通道参数，将不包含通道11的参数。设置应答报文为（正常状态），见表B.2。

表B.2 设置清空应答报文(正常状态)

| 字段名   | 值            | 长度(字节) | 描述                     |
|-------|--------------|--------|------------------------|
| 上位机ID | 0x12345678   | 4      | 视实际情况，与对应设置报文保持一致      |
| 设备ID  | 0x87654321   | 4      | 视实际情况，与对应设置报文保持一致      |
| 数据帧标识 | xxxx         | 2      | 视实际情况，与对应设置报文保持一致      |
| 时间戳   | xxxxxxxxxxxx | 6      | 视实际情况而定，实际回复报文时间       |
| 帧类型   | 0x21         | 1      | 表示设置数据响应               |
| 数据值数量 | 0x0001       | 2      | 该条报文操作1个数据值            |
| 数据值索引 | 0x0001       | 2      | 第1个数据值序号               |
| 标识号长度 | 0x04         | 1      | 该对象标识号长度为4             |
| 一级标识号 | 0x05         | 1      | 一级对象标识号5，对应：检测器信息      |
| 二级标识号 | 0x02         | 1      | 二级对象标识号2，对应：检测器配置表     |
| 三级标识号 | 0x00         | 1      | 三级对象标识号0，对应：检测器三级所有标识号 |
| 四级标识号 | 0x0B         | 1      | 通道11                   |
| 数据值长度 | 0x0001       | 2      | 第1个数据值的长度              |
| 数据值   | 0x00         | 1      | 设置正常                   |

设置回复报文为有错状态，见表B.3。

表B.3 设置回复报文(有错状态)

| 字段名   | 值            | 长度(字节) | 描述                     |
|-------|--------------|--------|------------------------|
| 上位机ID | 0x12345678   | 4      | 视实际情况，与对应设置报文保持一致      |
| 设备ID  | 0x87654321   | 4      | 视实际情况，与对应设置报文保持一致      |
| 数据帧标识 | xxxx         | 2      | 视实际情况，与对应设置报文保持一致      |
| 时间戳   | xxxxxxxxxxxx | 6      | 视实际情况而定，实际回复报文时间       |
| 帧类型   | 0x22         | 1      | 表示设置出错回复               |
| 数据值数量 | 0x0001       | 2      | 该条报文操作1个数据值            |
| 数据值索引 | 0x0001       | 2      | 第1个数据值序号               |
| 标识号长度 | 0x04         | 1      | 该对象标识号长度为4             |
| 一级标识号 | 0x05         | 1      | 一级对象标识号5，对应：检测器信息      |
| 二级标识号 | 0x02         | 1      | 二级对象标识号2，对应：检测器配置表     |
| 三级标识号 | 0x00         | 1      | 三级对象标识号0，对应：检测器三级所有标识号 |
| 四级标识号 | 0x0B         | 1      | 通道11                   |
| 数据值长度 | 0x0001       | 2      | 第1个数据值的长度              |
| 数据值   | 0x60         | 1      | 设置异常，无访问权限（示例）         |

## B.5 JSON 报文示例

一般报文格式参见总则相关章节。

本章节描述特殊类型的报文操作。

### B.5.1 设置类型示例1 – 设置相位

设置相位的配置信息示例：

设置2个相位，设置报文如下所示：

```
{
 "version": "1.00", // 版本号
 "authorizeMethod": 1, // MD5加密认证方式
 "user": "admin", // 用户名
 "code": "0123456789ABCDEF", // 认证码
 "type": 2, // 信号机协议标识
 "hostID": 12345678, // 上位机ID
 "devID": 87654321, // 设备ID
 "frameID": xx, // 根据实际情况
 "timestamp": xxxxxxxxxxxxxx, // 根据实际情况
 "frameType": "Type_Set", // 设置
 "data":
 {
 "phase":
 {
 "phaseConfigTable":
 [
 {
 "index": 1,
 "channelIncluded": [1],
 "onLoseStep1LightType": "Light_Status_GreenFlash",
 "onLoseStep1Time": 3,
 "onLoseStep2LightType": "Light_Status_Yellow",
 "onLoseStep2Time": 3,
 "onLoseStep3LightType": "Light_Status_Red",

```

```

 "onLoseStep3Time": 1,
 "onGetStep1LightType": "Light_Status_RedFlash",
 "onGetStep1Time": 3,
 "onGetStep2LightType": "Light_Status_Unavailable",
 "onGetStep2Time": 0,
 "onGetStep3LightType": "Light_Status_Unavailable",
 "onGetStep3Time": 0,
 "onStartupGetStep1LightType ": "Light_Status_YellowFlash",
 "onStartupGetStep1Time": 10,
 "onStartupGetStep2LightType ": "Light_Status_Red",
 "onStartupGetStep2Time": 5,
 "onStartupGetStep3LightType ": "Light_Status_Unavailable",
 "onStartupGetStep3Time": 0,
 "onStartupLoseStep1LightType ": "Light_Status_YellowFlash",
 "onStartupLoseStep1Time": 10,
 "onStartupLoseStep2LightType ": "Light_Status_Red",
 "onStartupLoseStep2Time": 5,
 "onStartupLoseStep3LightType ": "Light_Status_Unavailable",
 "onStartupLoseStep3Time": 0,
 "minGreen": 15,
 "maxGreen1": 30,
 "maxGreen2": 60,
 "extendedGreen": 3,
 "phaseRequest": [1],
 "maxWaste": 5,
 "lossTimeOnSwitch": 6
 },
 {
 "index": 2,
 "channelIncluded": [2],
 "onLoseStep1LightType": "Light_Status_GreenFlash",
 "onLoseStep1Time": 3,
 "onLoseStep2LightType": "Light_Status_Yellow",
 "onLoseStep2Time": 3,
 "onLoseStep3LightType": "Light_Status_Red",
 "onLoseStep3Time": 1,
 "onGetStep1LightType": "Light_Status_RedFlash",
 "onGetStep1Time": 3,
 "onGetStep2LightType": "Light_Status_Unavailable",
 "onGetStep2Time": 0,
 "onGetStep3LightType": "Light_Status_Unavailable",
 "onGetStep3Time": 0,
 "onStartupGetStep1LightType ": "Light_Status_YellowFlash",
 "onStartupGetStep1Time": 10,
 "onStartupGetStep2LightType ": "Light_Status_Red",
 "onStartupGetStep2Time": 5,
 "onStartupGetStep3LightType ": "Light_Status_Unavailable",
 "onStartupGetStep3Time": 0,
 "onStartupLoseStep1LightType ": "Light_Status_YellowFlash",
 "onStartupLoseStep1Time": 10,
 "onStartupLoseStep2LightType ": "Light_Status_Red",
 "onStartupLoseStep2Time": 5,
 "onStartupLoseStep3LightType ": "Light_Status_Unavailable",
 "onStartupLoseStep3Time": 0,
 "minGreen": 15,
 "maxGreen1": 30,
 "maxGreen2": 60,
 "extendedGreen": 3,
 "phaseRequest": [2],
 "maxWaste": 5,
 "lossTimeOnSwitch": 6
 }
]
}

```

```
 }
}
```

设置应答报文为全部“正常状态”，不需要填data内容，如下所示：

```
{
 "version": "1.00", // 版本号
 "authorizeMethod": 0, // 应答无需加密认证
 "type": 2, // 信号机协议标识
 "hostID": 12345678, // 上位机ID，与设置帧保持一致
 "devID": 87654321, // 设备ID，与设置帧保持一致
 "frameID": xx, // 根据实际情况，与设置帧保持一致
 "timestamp": xxxxxxxxxxxx, // 根据实际情况，实际回复时间
 "frameType": "Type_SetReply", // 设置应答
}
```

设置回复报文部分为“有错状态”，正确设置的标识号可以不必回复，只需要回复错误的标识号。例如设置的失去路权过渡灯色1类型不是预定义的类型，如下所示：

```
{
 "version": "1.00", // 版本号
 "authorizeMethod": 0, // 应答无需加密认证
 "type": 2, // 信号机协议标识
 "hostID": 12345678, // 上位机ID，与设置帧保持一致
 "devID": 87654321, // 设备ID，与设置帧保持一致
 "frameID": xx, // 根据实际情况，与设置帧保持一致
 "timestamp": xxxxxxxxxxxx, // 根据实际情况，实际回复时间
 "frameType": "Type_SetReply", // 设置应答
 "error":
 {
 "phase":
 {
 "phaseConfigTable":
 [
 {
 "index": 1,
 "onLoseStep1LightType": "Status_BadValue",
 // 相位 1 参数 OnLoseStep1LightType 错误码，视情况而定
 },
 {
 "index": 2,
 "onLoseStep1LightType": "Status_BadValue",
 // 相位 2 参数 OnLoseStep1LightType 错误码，视情况而定
 }
]
 }
 }
}
```

### B.5.2 设置类型示例2 – 清空对象

现要更新相位1，并清空相位2，报文如下所示：

```
{
 "version": "1.00", // 版本号
 "authorizeMethod": 1, // MD5加密认证方式
 "user": "admin", // 用户名
 "code": "0123456789ABCDEF", // 认证码
 "type": 2, // 信号机协议标识
 "hostID": 12345678, // 上位机ID
 "devID": 87654321, // 设备ID
 "frameID": xx, // 根据实际情况
 "timestamp": xxxxxxxxxxxx, // 根据实际情况
}
```

```

"frameType": "Type_Set", // 设置
"data":
{
 "phase":
 {
 "phaseConfigTable":
 [
 {
 "index": 1,
 "channelIncluded": [1],
 "onLoseStep1LightType": "Light_Status_GreenFlash",
 "onLoseStep1Time": 3,
 "onLoseStep2LightType": "Light_Status_Yellow",
 "onLoseStep2Time": 3,
 "onLoseStep3LightType": "Light_Status_Red",
 "onLoseStep3Time": 1,
 "onGetStep1LightType": "Light_Status_RedFlash",
 "onGetStep1Time": 3,
 "onGetStep2LightType": "Light_Status_Unavailable",
 "onGetStep2Time": 0,
 "onGetStep3LightType": "Light_Status_Unavailable",
 "onGetStep3Time": 0,
 "onStartupGetStep1LightType ": "Light_Status_YellowFlash",
 "onStartupGetStep1Time": 10,
 "onStartupGetStep2LightType ": "Light_Status_Red",
 "onStartupGetStep2Time": 5,
 "onStartupGetStep3LightType ": "Light_Status_Unavailable",
 "onStartupGetStep3Time": 0,
 "onStartupLoseStep1LightType ": "Light_Status_YellowFlash",
 "onStartupLoseStep1Time": 10,
 "onStartupLoseStep2LightType ": "Light_Status_Red",
 "onStartupLoseStep2Time": 5,
 "onStartupLoseStep3LightType ": "Light_Status_Unavailable",
 "onStartupLoseStep3Time": 0,
 "minGreen": 15,
 "maxGreen1": 30,
 "maxGreen2": 60,
 "extendedGreen": 3,
 "phaseRequest": [1],
 "maxWaste": 5,
 "lossTimeOnSwitch": 6
 },
 {
 "index": 2,
 "allElement": 0, // allElement 代表所有元素
 }
]
 }
}

```

清空成功后，再次查询相位参数，相位2所有元素是清空状态。

## 参 考 文 献

- [1] GB 25280—2016 道路交通信号控制机
- [2] GB/T 20999—2017 交通信号控制机与上位机间的数据通信协议
- [3] SZZD X—2018 深圳信号机通信协议标准
- [4] GB 14886—2016 道路交通信号灯设置与安装规范
- [5] GA/T 920—2010 道路交通信号控制机与车辆检测器间的通信协议
- [6] GA/T 1049.2—2013 公安交通集成指挥平台通信协议第2部分交通信号控制系统
- [7] GA/T XXXX—XXXX 道路交通车路协同信息服务通用技术要求
- [8] GA/T 1743—2020 道路交通信号控制机信息发布接口规范
- [9] GB/T 43229—2023 交通信号控制机与车辆检测器之间通信协议
- [10] GA/T 920-2010 道路交通信号控制机与车辆检测器间的通信协议
- [11] 美国国家智能运输系统通信协议(NTCIP)
- [12] Signal Timing Manual - Second Edition, TRB's National Cooperative Highway Research Program (NCHRP) Report 812
- [13] SAE J2735 V2X Communications Message Set Dictionary

团体标准  
《基于物联网的道路交通安全设备通信协议  
第 2 部分：交通信号控制机》

（征求意见稿）

编

制

说

明

标准起草工作组

2025 年 7 月

# 《基于物联网的道路交通安全设备通信协议 第2部分：交通信号控制机》编制说明

## 一、工作简况

### （一）任务来源

#### （1）项目编号

为积极贯彻落实《数字交通“十四五”发展规划》对于智能交通的部署要求，提高交通管理效率，促进智能交通发展，公安部道路交通安全研究中心牵头，联合地方公安交管部门、相关高校及科技公司，引入通信领域的软件定义网络理念，制定基于物联网的道路交通安全设备通信协议。

根据中国道路交通安全协会《关于同意《基于物联网的道路交通安全设备通信协议第2部分：交通信号控制机》团体标准立项的通知》（中交安协通〔2023〕31号）的要求，由苏州科达科技股份有限公司负责本标准的制定工作。

#### （2）制定背景

随着交通行业的快速发展，交通设备的种类和数量不断增加，包括交通信号控制机、可变交通标志、交通雷达、监控摄像机等交通设备在城市交通管理中发挥着重要作用。长久以来，道路交通安全设备制造厂商都在发展或采用不同的、私有的数据通信协议，给同一系统中不同厂商设备或者不同系统中设备间的协同控制造成困扰，设备和管理平台之间无法快速、有效地进行信息交流和协同工作，跨设备的互联互通局限性较大，从而造成了交通管理效率低下和资源浪费的问题。

本系列标准目的是提供各类交通安全设备的统一协议，供所有交通安全设备生产厂商和上层应用系统开发厂商设计和开发通信接口使用，从而提高交通管理效率，降低设备维护成本。

#### （3）预研情况

2023年4月，组织开展标准立项的前期预研工作；

2023年8月，成立标准编制技术工作组，并组织开工作组启动会议，制定标准内容编制方案。

2023年4月-2023年11月，标准编制人员实地调研北京，天津，南通等地以及相关运输企业对于道路交通安全设备应用现状，各地关于道路交通安全设备的通信协议进行归纳整理。

2023年11月，标准编制工作组召开标准专家咨询会，明确了本系列标准名为《基于物联网的道路交通安全设备通信协议》，确定了总则作为系列标准的第1部分。

2023年11月，标准编制工作组制定系列标准大纲，并经过多次研究和讨论，整理调研数据、梳理收集的资料，按照研究大纲起草《基于物联网的道路交通安全设备通信协议 第1部分：总则》草案。

### （二）主要工作过程

主要阶段包括：

#### 起草阶段（18个月）：

2023年11月-12月，标准编制工作组制定系列标准大纲，并经过多次研究和讨论，整理调研数据、梳理收集的资料，按照研究大纲起草《基于物联网的道路交通安全设备通信协议 第2部分：

交通信号控制机》草案。

2023 年 12 月，向中国道路交通安全协会申请团体标准立项。

2024 年 1 月-6 月，标准编制工作组对并经过多次会议研讨和和交流，对《基于物联网的道路交通安全设备通信协议 第 3 部分:LED 可变信息标志》草案进行编制。

2024 年 7 月，召开标准专家咨询会，对标准定位、架构、内容等进行讨论，并针对专家意见进行修改。

2024 年 9 月，召开系列标准专家研讨会，重点对《基于物联网的道路交通安全设备通信协议 第 1 部分：总则》的标准格式和覆盖内容及撰写形式进行研讨。

2024 年 10 月-2025 年 5 月，根据《第 1 部分：总则》的修改情况，同步开展《第 3 部分：交通信号控制机》的标准草案修订。

2025 年 6 月召开系列标准专家研讨会，会后根据专家意见修改，进一步完善了标准草案。

2025 年 8 月，标准编制工作组形成了《基于物联网的道路交通安全设备通信协议 第 3 部分：LED 道路交通诱导可变信息标志》标准征求意见稿和编制说明，并上报中国道路交通安全协会。

**征求意见阶段：**计划于 2025 年 9 月完成向有关单位和专家征求意见，并召开征求意见稿讨论会，形成征求意见表、征求意见汇总处理表等上报协会。

**审查阶段：**计划于 2025 年 10 月前完成“征求意见稿”修改，标准编制组将召开标准咨询会议，对标准进一步完善，形成“送审稿”及条文说明，报送协会提请审查会专家审查。

**报批阶段：**计划于 2025 年 12 月前完成“送审稿”修改，对审定意见逐条进行了梳理，并对团标准文本中相关内容的表述进行了调整，形成报批稿及条文说明上报，待审批。

### （三）标准主要起草单位基本情况介绍及其所做的工作

本标准主要起草单位为苏州科达科技股份有限公司、公安部道路交通安全研究中心、南通市公安局交通警察支队、天津光电比特信息技术有限公司、北京航空航天大学、北京四维图新科技股份有限公司。

**苏州科达科技股份有限公司**，引领视讯与安防。1995-2020 年，科达在竞争激烈的视频信息化领域稳健成长。近年来，在人工智能技术战略及行业深耕业务战略的双重推动下，科达智能交通业务取得了更快速的发展，截止目前，科达在全国已交付超过 600 个智能交通项目，获得 2018AI+智慧交通十大优秀企业、2020 中国智能交通建设推荐品牌等荣誉。

- （1）自主研发 AI 超微光等前沿创新前端产品
- （2）平台业务覆盖情指勤督、精准打击等多项实战场景
- （3）拥有智慧检查站、服务区等精细化、专业化解决方案
- （4）发布开源信控系统，从源头变革交通治理

**公安部道路交通安全研究中心**（以下简称“道研中心”）系公安部在京直属科研事业单位，2011 年 9 月经中央编办批准成立。主要承担全国道路交通管理规划研究、交通法规标准研究、机动车辆和驾驶人安全研究、道路安全研究、宣传教育研究、数据统计及相关信息化建设等工作。道研中心联合清华大学、同济大学、东南大学等研究机构建立了多个交通安全实验室，联合江苏省公安厅交通警察总队、四川省公安厅交通警察总队、湖北省公安厅交通警察总队、深圳市公安局交通警察局建立了公安交通管理执法规范化建设研究基地，与美国、欧洲、日本等 10 多个发达国家、地区相关机构和组织建立了良好的交流合作关系。自成立以来，道研中心共承担了 18 项国家级项目，主持和参与了 25 项省部级科研项目、136 项中央财政资金支持项目的攻关工作，获得公安部科学技术奖一等奖 1 项、二等奖 1 项、三等奖 2 项，参与编制国家标准 4 项、主持行业标准 20 项，先后获得国家专利 20 余项，软件著作权 50 余项，出版专著 42 部，发表学术文章 400 余篇。各项专题研究成果得到公安部领导认可，并部署全国公安机关交通管理部门推广应用。

**南通市公安局交通警察支队**是南通市公安局内设的交通管理部门，下辖 4 个科室、3 个专业大队、17 个勤务大队，主要负责组织、指导全市道路交通安全管理工作，具体职能包括交通秩序管

理、交通事故处理、交通安全宣传、交通安全保卫、智能交通建设、高速公路交通治安管理及车辆、驾驶人管理等。近年来，交警支队指导和带领全市公安机关交通管理部门和全体交警、辅警在市委、市政府和市公安局党委的正确领导下，主动融入打造“升级版南通公安”总体布局，全面深化平安交通、文明交通、民生交通和职业警队建设，强势推进道路交通安全管理社会化进程，全力服务群众出行，全市道路交通安全管理水平稳步提升，交通秩序明显好转、交通环境显著改善、交通安全形势持续平稳，受到社会各界和各级领导的广泛赞誉。农村道路交通安全管理、智能交通建设、源头监管、执法规范化、交通事故处理服务、实战化警队建设、车辆驾驶人管理、交通肇事逃逸案件常态侦破机制等工作走在全省乃至全国前列。支队以“特别能战斗，特别能吃苦，特别能奉献，特别能忍耐”为队伍特质，呼之即来，来之能战，战之能胜，先后数十次被公安部、省厅表彰，多次立功受奖，执法质量始终保持在全省优秀行列，作风建设始终位居全市行政执法单位第一名，车管所被评为“全国公安优秀基层单位”、“全国一等车管所”，队伍中涌现出一大批全国、全省先进典型。

**天津光电比特信息技术有限公司**成立于 2007 年，注册资本 5000 万元，是一家专注于开发智慧交通设备的高新技术企业。公司主要研发，生产和销售交通诱导设备、LED 显示设备、收费机电设备、绿色供电系统、通信传输系统、智能交通设备，以及收费站机电设备和通信设备，并提供系统服务。公司通过了 GB/T19001-2016 质量管理体系、GB/T 24001-2016 环境管理体系 GB/T45001-2020 职业健康安全管理体系、GB/T22080-2016 信息安全管理体系第三方认证。凭借先进的研发技术、严格的品质管理、可靠的生产工艺和完善的服务体系，确保产品品质和服务品质。公司依靠雄厚的技术实力和深厚的文化底蕴，以一流的产品品质和优质的售前、售后服务，满足客户的全方位需求，为客户创造超越期望的价值，达到客户、供应商、销售商和制造商的共赢。公司拥有一支有着 10 多年研发经验的技术研发团队，有一支高效的技术型销售团队和技术服务团队。公司凭借行业内资深专业人才的优势，倾力打造高效、快速响应的运营团队，致力成为智能交通设备行业内最具竞争力的企业。

**北京航空航天大学**（Beihang University）简称“北航”，位于北京市，是中华人民共和国工业和信息化部直属的全国重点大学，中央直管高校，位列国家“双一流”、“985 工程”、“211 工程”重点建设高校，入选珠峰计划、2011 计划、111 计划、卓越工程师教育培养计划、国家建设高水平大学公派研究生项目、中国政府奖学金来华留学生接收院校、国家级新工科研究与实践项目、国家级大学生创新创业训练计划、国家大学生创新性实验计划、全国深化创新创业教育改革示范高校、强基计划试点高校。

**北京四维图新科技股份有限公司**（简称四维图新），是中国领先的导航地图、动态交通信息及汽车综合信息服务提供商，致力于为全球客户提供专业化高品质的电子地图数据产品和服务，主要业务包括地图导航、高级辅助驾驶及自动驾驶、车规级芯片、位置大数据平台和车联网。公司是全球第三家、中国第一家通过 TS16949（国际汽车工业质量管理体系）认证的地图厂商。公司总部位于北京海淀，并在武汉、西安、上海、合肥、台湾等地设立了多个研发中心和数据生产中心。目前总共设有 23 个分子公司，5 个研发事业部。四维图新已于 2019 年在安徽合肥成立专注于自动驾驶地图生产的第二总部。四维图新聚焦汽车智能化主赛道，发展智云、智驾、智舱、智芯四大解决方案，同时聚焦智能交通、数字孪生城市解决方案，将多年积淀的数据与技术产品化、平台化，借助行业趋势及生态优势，不断在深耕领域拓展创新。2022 年，公司产品不断获得国内外客户的认可，产品量产交付进一步加快，收入同比 2021 年实现增长。其中，智云业务陆续落地重要客户，营收增长 3.14%；智能驾驶软硬一体产品签订量产订单并陆续出货，智驾业务实现收入 1.3 亿元，收入增长 20 倍；智芯业务主要得益于车规级微控制器芯片（MCU）、智能座舱芯片（SoC）持续放量，实现收入逾 5 亿元，增长 40%以上。

（四）起草人及其所做的工作

| 主要参加单位 | 成员 | 主要工作 |
|--------|----|------|
|--------|----|------|

|                |                     |                                           |
|----------------|---------------------|-------------------------------------------|
| 苏州科达科技股份有限公司   | 蒋松涛、章文超、刘黎明         | 负责道路信号机技术路线设计及编写                          |
| 公安部道路交通安全研究中心  | 胡伟超、于鹏程、刘雨桐、李小松、张博越 | 负责标准编制工作总体把关、组织协调、实地调研、标准文本格式编制等          |
| 南通市公安局交通警察支队   | 顾坚，范荣建，张建才，许世俊      | 负责设备安全性要求等部分正文的编写、论证                      |
| 天津光电比特信息技术有限公司 | 刘志合、曹玉芳、贾雪冬         | 负责道路 LED 可变信息标志、交通状态检测器、激光雷达等设备的技术设计和正文编写 |
| 北京航空航天大学       | 于海洋、任毅龙、常屹卓、陈明      | 负责资料查询、方法验证、通信协议的编写                       |
| 北京四维图新科技股份有限公司 | 刘士宽、李博，孙伟           | 负责可编程交通系统高精地图接口规范的开发                      |

## 二、编制原则

**1、原则性。**标准的编制严格遵守《中华人民共和国标准法》、《中华人民共和国标准法实施细则》、GB / T 1.1-2020《标准化工作导则 第1部分：标准的结构和起草规则》及相关法规的要求进行。

**2、适应性。**本标准制定适用于面向智能网联的可编程交通管控通讯协议，顺应当前智能交通网络管控的理念和趋势，适应地方交通安全风险防范需求，并注重与其他交通设备的统一性和协调性。

**3、先进性。**根据公安交管部门实现交通智能管控的要求，该标准对提高交通网络的统一调配效率具有重要意义。

## 三、标准内容的起草

### （一）主要技术内容的确定和依据

（1）关于“前言”。本部分规定了《基于物联网的道路交通安全设备通信协议》系列标准结构，该系列标准共包括 11 部分，本文件为该系列标准的第 2 部分：交通信号控制机。

（2）关于“1 范围”。本部分规定了本文件范围为交通信号控制机与集中控制器或上位机进行数据通信的数据帧格式内容，适用于物联网条件下道路交通安全设备中交通信号控制机与集中控制器或上位机间的数据通信。

（3）关于“2 规范性引用文件”。本部分对文中出现的规范性引用文件进行了罗列。

（4）关于“3 术语和定义”。本部分对本文件首次出现的“交通信号控制机”等术语进行了定义，同时对本文件中出现的缩略语进行了说明。

（5）关于“4 技术要求”。

① 对于“4.1 一般规定”。可参照系列标准第 1 部分：总则。

② 对于“4.2 数据帧内容”。明确了本文件涉及的“交通信号控制机”的数据帧标识号及数据值内容。例如，通用数据的标识号为 1. x，基础信息标识号为 2. x 等，并以表格的形式对不同数据类的数据内容进一步进行明确。

③ 对于“4.3 数据值 MIB”。根据 ASN.1 语言规范，给出了所有数据信息的 ASN.1 描述示例，作为该设备的数据字典以支持 json 与 xml 等数据格式的协议开发。

(6) 对于“附录 A 通信协议规范定义”。规定了用户权限、灯组状态、信号机运行模式等规范性定义，并说明了数据值定义可扩展，各厂家可以根据需求定义。

(7) 对于“附录 B 通信规程及报文示例”。规定了信号机与上位机采用总则定义的协议或者 SNMP v3 协议实现通信交互。并说明了上报类型、设置类型的 json 格式报文。

## (二) 标准中英文内容的汉译英情况

标准英文译名为: Road traffic safety equipment communication protocol based on Internet of Things Part 2: Traffic signal controller。

## 四、试验验证结果及分析

暂无。

## 五、标准水平和预期效益

1. 促进智慧交通发展：统一通信协议为智慧交通建设提供了基础。智慧交通系统通过统一的协议可以快速、有效的实现各种交通设备之间实现信息共享和协同工作。

2. 促进技术创新和产业升级：通过统一交通设备的通信协议，可以促进技术创新和产业升级。不同厂商的交通设备可以基于同一通信协议进行开发，降低了技术研发和生产的成本。同时，统一的通信协议也为交通设备的互联互通提供了基础，促进了各类交通设备和智能交通系统的融合与发展。

3. 提升产业竞争力：通过统一交通设备的通信协议，可以提升产业竞争力。统一的通信协议可以降低设备开发和维护成本，提高设备的互操作性和兼容性，加快设备的市场推广和应用。这将使得企业能够更快速地推出新产品，提高产品质量和性能，增强在市场竞争中的竞争力。

4. 促进行业合作与协同发展：通过统一交通设备的通信协议，可以促进行业内企业之间的合作与协同发展。一致的通信协议将使得不同厂商的设备能够互相配合，实现信息共享和协同工作，提高整体效能。这将有助于激发行业内的合作潜力，加强行业内各方之间的合作与交流，推动行业的快速发展。

## 六、采用国际标准和国外先进标准情况

该系列标准参考了美国 NTCIP(National Transportation Communications for ITS Protocol) 标准通讯协议的架构，该协议基于 SNMP 架构，交通语义采用环概念。NTCIP 能够确保交通控制与智能运输系统(ITS) 组成单元彼此之间的“互操作性”与“互换性”。现行工业标准中 SNMP 是最广泛使用的网络设备管理协议标准，本次协议拟定中，引入 SNMP 管理要求，以便实现统一的设备管控。

## 七、与有关法律、行政法规及相关标准的关系

国内与本协议相关的标准有 GB25280 - 2016《道路交通信号控制机》中附录 A“指令和消息格式”以及 GB/T 20999-2017《交通信号控制机与上位机间的数据通信协议》。其中 GB25280-2016 版本中的指令和消息格式描述粗略，实际上很难应用于协议对接，GB/T 20999-2017 定义相对完整，目前多家企业已经实现主要基于该标准的协议对接，但是由于原始定义内容仍然有缺失，因此大部分场合都是对此协议进行增补，导致实际上的不兼容。

交通信号控制机的通信协议当前主要依据 GB25280 - 2016《道路交通信号控制机》中附录 A“指令和消息格式”以及 GB/T 20999-2017《交通信号控制机与上位机间的数据通信协议》，其中 GB25280-2016 版本中的指令和消息格式描述粗略，实际上很难应用于协议对接，GB/T 20999-2017

定义相对完整，目前多家企业已经实现主要基于该标准的协议对接，但是由于原始定义内容仍然有缺失，因此大部分场合都是对此协议进行增补，导致实际上的不兼容。

目前相关协议中对交通语义的定义都是采用阶段概念，而国际上主要存在阶段概念以及环概念这两种交通语义体系，各有优缺点。国内部分交通信号企业已经实现环概念并在此基础上开发了灵活的控制逻辑。因此在现有标准上引入环概念有助于实现更广泛的环境适应性。

#### **八、重大分歧意见的处理过程和依据**

本标准未产生重大分歧意见。

#### **九、标准性质的建议**

无。

#### **十、贯彻、实施标准的要求和建议**

本标准为首次发布，自愿采用。

#### **十一、废止、替代现行有关标准的建议**

无。

#### **十二、其他予以说明的事项**

无。