

OCIT[®]

Open Communication Interface for Road Traffic Control Systems

Offene Schnittstellen für die Straßenverkehrstechnik

OCIT-Outstations Lichtsignalsteuergeräte

OCIT-O_Lstg_V2.0_A04

OCIT Developer Group (ODG)

OCIT[®] ist eine registrierte Marke der Firmen Dambach, Siemens, Signalbau Huber, Stoye und Stührenberg

OCIT-Outstations

Lichtsignalsteuergeräte

Dokument: OCIT-O_Lstg_V2.0_A04

Herausgeber: OCIT Developer Group (ODG)

Kontakt: www.ocit.org

Copyright © 2012 ODG. Änderungen vorbehalten. Dokumente mit Versions- oder Ausgabestände neueren Datums ersetzen alle Inhalte vorhergehender Versionen.

Inhaltsverzeichnis

Spezifikationen	14
1 Einführung	14
1.1 Unterstützte Funktionen	14
1.2 Unterstützung von OCIT-I Schnittstellen	15
2 Geräte- und Systemfunktionen	16
2.1 Lichtsignalsteuergeräte mit OCIT-O Schnittstelle.....	16
2.2 Fernversorgung von Lichtsignalsteuergeräten.....	17
2.3 Versorgungswege.....	17
2.3.1 Versorgungsdaten.....	19
2.3.2 Versionierung.....	21
2.3.3 Anforderungen an den Versorgungsdatenserver	21
2.3.4 Anforderungen an die Lichtsignalsteuergeräte.....	22
2.4 Übertragungsgeschwindigkeit	22
2.5 Synchronisierung u. Rückrechenverfahren	23
2.5.1 Berechnungsbeispiele für Rückrechenverfahren	24
2.6 Teilknoten	25
2.7 Relative Knoten (mehrere Knoten innerhalb eines Lichtsignalsteuergerätes)	25
2.8 Zeitstempel.....	26
2.9 Zeitzählung.....	26
2.9.1 Zeitschaltwerte und Umlauf.....	26
2.10 Vorgangskennung.....	27
3 Objektdefinitionen	30
3.1 Übertragung von Versorgungsdaten	30
3.1.1 Objekt Transaction	30
3.1.2 Objekt SupplyTransaction (Versorgungstransaktion).....	39
3.1.3 Objekt TransferParameterBlock.....	44

3.2	Versionierung der Versorgungsdaten	47
3.2.1	Standard-Verfahren zur Checksummenbildung	53
3.2.2	Objekt Versionierung	57
3.2.2.1	Enum VDArt 1: 680	57
3.2.2.2	Objekt VDVersion	57
3.2.2.3	LsaVersion	59
3.2.2.4	LsaVersionPlus	61
3.2.2.5	GesamtVersion	63
3.3	Versorgungsobjekte	65
3.3.1	Objekt VersorgbaresObjekt	65
3.3.2	Block 1: Verkehrstechnische Grunddaten / Festzeit	67
3.3.2.1	Objekt Eprogramm	67
3.3.2.2	Objekt AProgramm	68
3.3.2.3	Objekt SignalprogrammV	69
3.3.2.3.1	OCIT-O Referenzen auf Zusatzübergänge	70
3.3.2.4	Objekt Versatzzeitenmatrix	71
3.3.2.4.1	Versatzvarianten (Beispiele)	74
3.3.2.5	Objekt VTZwischenzeitenmatrix	77
3.3.2.6	Objekt VTMinFreigabe	78
3.3.2.7	Objekt VTMinGesperrt	79
3.3.3	Block 2: Daten mit Netzbezug	80
3.3.3.1	Objekt Kopfdaten	80
3.3.3.2	Schaltuhr	81
3.3.3.2.1	Projektspezifische Modifikationen der Schaltuhr	81
3.3.3.2.2	Objekt Tagesplan	81
3.3.3.2.3	Objekt Wochenplan	83
3.3.3.2.4	Objekt SondertagJaehrlich	83

3.3.3.2.5	Objekt SondertagAufzaehlung.....	86
3.3.3.2.6	Objekt Zeitbereich.....	86
3.3.4	Block 3: VA-Steuerverfahren	88
3.3.4.1	Objekt BinaerVASteuerverfahren	88
3.3.5	Block 4: VA-Parameter	89
3.3.5.1	Objekt BinaerVAParameter	89
3.4	Zentrale Schaltwünsche.....	90
3.4.1	OCIT-O konforme Konstellationen der zentralen Schaltwünsche	92
3.4.2	Struktur ZEITINTERVALL.....	95
3.4.3	Typen und Pfade.....	96
3.4.4	Objekt ZSignalProgramm.....	97
3.4.5	Objekt ZKnotenEinAus	98
3.4.6	Objekt ZTeilKnoten.....	101
3.4.7	Objekt ZSondereingriff.....	103
3.4.8	Signalprogramm Modifikationen.....	105
3.4.8.1	Objekt ZModEinAus.....	106
3.4.8.2	Objekt IModEinAus.....	107
3.4.8.3	Objekt ZVAEinAus	108
3.4.8.4	Objekt ZOepnvEinAus.....	108
3.4.8.4.1	ObjektZVAIndividualverkehrEinAus.....	110
3.4.8.5	Kombination von Modifikationen	110
3.4.9	Projektspezifische Modifikationen	111
3.4.9.1	Objekt ZProjEinAus	112
3.4.9.2	Objekt IProjEinAus	113
3.4.10	Objekt ZentralenSchaltwunsch	114
3.4.11	Objekt ISignalProgramm.....	117
3.4.12	Objekt IKnotenEinAus	117

3.4.13	Objekt ITeilknoten	118
3.4.14	Objekt ISondereingriff.....	118
3.4.15	Objekt IVAEinAus	119
3.4.16	Objekt IVAIndividualverkehrEinAus	119
3.4.17	Objekt IOepnvEinAus	120
3.4.18	Objekt IBetriebsart	120
3.4.19	Objekt IstVektor	121
3.4.20	Objekt Gerätestatus	123
3.5	Meldungen und Messwerte.....	124
3.5.1	Objekttypen und Klassenübersicht.....	125
3.5.2	Messwertaufträge für Lichtsignalanlagen	127
3.5.2.1	Zyklisch abgefragter Auftrag (AuftragZykl).....	127
3.5.2.2	Auftrag bei Abtaständerungen (MWAuftragAbtastAenderung).....	128
3.5.2.3	Auftrag Abtaständerungen mit Wertevergleich (MWAuftragVergleich)...	129
3.5.2.4	Auftrag für asynchrone Prozessvariable (MWAuftragExtern)	130
3.5.2.5	Auftrag für Einzelschleifen-Erfassung (MWAuftragAbtastAB)	131
3.5.2.6	Auftrag für erweiterte Detektorwerte (MWAuftragDetExt)	132
3.5.2.7	Auftrag für R09-Telegramme (MWAuftragR09).....	133
3.5.2.8	Auftrag für erweiterte R09-Telegramme (ÖPNV-Eintrag AMLi).....	134
3.5.3	Auftragselemente.....	138
3.5.3.1	Auftragselement für binäre Eingänge (AEBinaer).....	138
3.5.3.2	Aggregierte Werte für binäre Eingänge (AEAggregiert)	139
3.5.3.3	Auftragselement für Anwenderprogrammwert (AEAPWert).....	141
3.5.3.4	Auftragselement blockweises Lesen von AP-Werten (AEAPWertVektor)	141
3.5.3.5	Auftragselement für Detektoren mit Zusatzinformationen (AEDetExt)	143
3.5.3.6	Auftragselement für erweiterte aggr. Det.-Werte (AEAggregiertExt)	144
3.5.3.7	Auftragselement für Visualisierungsdaten (AESiplOnline)	147

3.5.3.8	Auftragselement Signalbild (AESignalBild)	149
3.5.3.8.1	Signalbildcodierung	150
3.5.3.9	Auftragselement DigAusgang (AEDigAusgang).....	151
3.5.3.9.1	Codierung	152
3.5.3.10	Kombinationen von Aufträgen und Auftragselementen	152
3.5.4	AP-Werte	153
3.5.4.1	APWert (1:505)	154
3.5.4.1.1	APWertUshort (1:506)	155
3.5.4.1.2	APWertLong (1:507)	155
3.5.4.2	APWertRk (1:510).....	155
3.5.4.2.1	APWertRkUshort (1:511).....	155
3.5.4.2.2	APWertRkLong (1:512)	155
3.5.4.3	APWertBlock (1:508)	155
3.5.4.3.1	APWertRkBlock (1:513)	156
3.5.4.4	APWertGroup (1:515)	156
3.5.4.4.1	APWertGroupRk (1:516)	158
3.5.4.5	Standardisierte AP-Werte	159
3.5.4.6	Verfahrensparameter	159
3.5.5	Detektoren und Signale	160
3.5.5.1	Digitaler Eingang (DigEingang).....	160
3.5.5.2	Signalgruppe (SignalGruppe)	161
3.5.5.3	Signalgeber (SignalGeber).....	161
3.5.5.4	Signalkammer (SignalKammer)	162
3.5.5.5	Digitaler Ausgang (DigAusgang)	162
3.5.6	Archive der Lichtsignalsteuergeräte.....	163
3.5.6.1	Elementbeschreibungen Meldungsarchiv	165
3.5.6.1.1	Exclude Liste des Standard-Meldearchivs	168

3.5.6.1.2	Include Liste des Versorgungsarchivs	168
3.5.6.2	Elementbeschreibungen Betriebszustandsarchiv.....	168
3.5.6.3	Eigenschaften der Listen.....	170
Glossar	172

Dokumentenstand

Version Zustand	Verteiler	Datum	Kommentar
V2.0 A01	PUBLIC	20. 03.08	<p>Änderungen in:</p> <ul style="list-style-type: none"> 1 Einführung 2 Geräte- und Systemfunktionen <ul style="list-style-type: none"> 2.2 Fernversorgung von Lichtsignalsteuergeräten 2.3 Versorgungswege 2.5 Synchronisierung und Rückrechenverfahren 2.6 Teilknoten 2.9 Zeitählung 2.10 Vorgangskennung 3 Objektdefinitionen <ul style="list-style-type: none"> 3.1 Übertragung von Versorgungsdaten 3.2 Versionierung der Versorgungsdaten 3.3 Versorgungsobjekte <ul style="list-style-type: none"> 3.4.1 OCIT-O konforme Konstellationen der zentralen Schaltwünsche <ul style="list-style-type: none"> 3.4.18 Objekt IBetriebsart 3.4.19 Objekt IstVektor 3.4.9 Projektspezifische Modifikationen 3.5.1 Objekttypen und Klassenübersicht <ul style="list-style-type: none"> 3.5.2.4 Auftrag für asynchrone Prozessvariable 3.5.2.6 Auftrag für erweiterte Detektorwerte 3.5.2.8 Auftrag für erweiterte R09-Telegramme 3.5.4.3 APWert Block <ul style="list-style-type: none"> 3.5.3.4 Auftragsэлеment für blockweises Lesen von AP-Werten (AEAPWertVektor) 3.5.3.5 Auftragsэлеment für Detektoren mit Zusatzinformationen 3.5.3.6 Auftragsэлеment für erweiterte aggregierte Det.-Werte 3.5.3.7 Auftragsэлеment für Visualisierungsdaten 3.5.4 AP-Werte 3.5.5 Detektoren und Signale 3.5.6 Archive der Lichtsignalsteuergeräte <ul style="list-style-type: none"> 3.5.6.1 Elementbeschreibung Meldungsarchiv <ul style="list-style-type: none"> 3.5.6.1.1 Exclude Liste des Standard-Meldearchivs 3.5.6.1.2 Include Liste des Versorgungsarchivs 3.5.6.2 Elementbeschreibung Betriebszustandsarchiv 3.5.6.3 Eigenschaften der Listen
V2.0 A02	PUBLIC	10.07.09	<ul style="list-style-type: none"> 1.1 Unterstützte Funktionen: Hinweis zur Kompatibilität eingefügt. 2.3.1 Versorgungsdaten: Hinweis zur Sorgfaltspflicht eingefügt. 2.3.4 und 2.9.1: Korrektur von Zeitwerten durch das Lichtsignalsteuergerät verboten. 2.6 Teilknoten: „TK(0)Keiner“ entfernt. Festlegung dazu

Version Zustand	Verteiler	Datum	Kommentar
			<p>präzisiert.</p> <p>2.10 Vorgangskennung: Erweiterung um Untertypen.</p> <p>3.1.1 Objekt Transaktion: Methode 100; Methode 104: Parameter_Invalid gelöscht, Methode 106: ObjectNotInBlock gelöscht, Text erweitert bei Otype 60306, Meldung spezifiziert bei Otype 60306; Entfernt wegen Vorschrift in RetCode SUPPLY_CHANGED bei Transaction.Check(), Parameter RelaxedMode von Transaction.Check(),Meldung SupplyModification.</p> <p>3.1.2 SupplyTransaction: Festlegung zur Zusatzmeldung bei SupplyVersionChanged geändert; Methode 120 und 121 korrigiert.</p> <p>3.2.1 Standard-Verfahren zur Checksummenbildung: Vorschrift präzisiert.</p> <p>3.2.2.2 Objekt VDversion: Pfad korrigiert; Hinweis zur herstellerspezifischen Änderung der Anwendersversorgung eingefügt; Wert NULLVALUA korrigiert.</p> <p>3.2.2.3 LsaVersion: Pfad korrigiert, Text in Origin korrigiert</p> <p>.</p> <p>3.3 Versorgungsobjekte: Text korrigiert.</p> <p>3.3.2.1 Objekt EProgramm.</p> <p>3.3.2.2 Objekt AProgramm.</p> <p>3.3.2.3 Objekt SignalprogrammV: Text korrigiert; VZ Matrix korrigiert.</p> <p>3.3.2.4 Versatzzeitenmatrix: Verkehrstechnische Bedeutung eingefügt.</p> <p>3.3.2.5 Objekt VTZwischenzeitmatrix.</p> <p>3.3.2.6 Objekt VTMinFreigabe.</p> <p>3.3.2.7 Objekt VTMinGesperrt).</p> <p>3.3.3.1 Objekt Kopfdaten: Wertebereich der UnitNr festgelegt.</p> <p>3.3.3.2.1 Projektspezifische Modifikationen der Schaltuhr: Priorität festgelegt.</p> <p>3.3.3.2.2 Objekt Tagesplan: Hinweis zum Eintrag der schaltwünsche eingefügt.</p> <p>3.3.3.2.3 Objekt Wochenplan: Hinweis zum Eintrag der schaltwünsche eingefügt.</p> <p>3.3.3.2.4 Objekt SondertagJaehrlich: Hinweis zur Priorität eingefügt.</p> <p>3.3.3.2.5 Objekt SondertagAufzaehlung: Hinweis zur Priorität eingefügt.</p> <p>3.3.3.2.6 Objekt Zeitbereich: Hinweis zur Priorität eingefügt.</p> <p>3.3.4.1 Objekt BinaerVASteuerverfahren: Angaben zur Kennzeichnung und Encoding eingefügt.</p> <p>3.3.5.1 Objekt BinaerVAParameter: Angaben zur Kennzeichnung und Encoding eingefügt.</p> <p>3.4 Zentrale Schaltwünsche: Hinweis Startzeit eigefügt.</p>

Version Zustand	Verteiler	Datum	Kommentar
			<p>3.4.1 OCIT-O konforme Konstellationen der zentralen Schaltwünsche: IstVektor in Tabelle der Schaltzustände eingefügt.</p> <p>3.4.2 Struktur Zeitintervall: Text geändert.</p> <p>3.4.8 Signalprogramm Modifikationen: Priorität der Schaltquelle festgelegt.</p> <p>3.4.8.1 Objekt ZModEinAus: Otype korrigiert.</p> <p>3.4.8.3 Objekt ZVAEinAus: Otype korrigiert.</p> <p>3.4.10 ZentralenSchaltwunsch: Methode „Schalte Knoten“ eingefügt.</p> <p>3.4.14 Objekt ISondereingriff: Name geändert von ISonderEingriff auf ISondereingriff; ID geändert auf 1:229.</p> <p>3.5.2.5 Auftrag für Einzelschleifen: Erklärung angefügt.</p> <p>3.5.2.8 Auftrag für erweiterte R09-Telegramme: Definition Fahrplanabweichung in Sek.</p> <p>3.5.3.1 Auftragsemet für binäre Eingänge: Text in SetChannel (DigEingang) korrigiert.</p> <p>3.5.3.2 Aggregierte Werte für binäre Eingänge: Text in AEAggregiert korrigiert.</p> <p>3.5.3.4 Auftragselement blockweises Lesen von AP-Werten. (AEAPWertVektor): Hinweis zum Hashwert eingefügt bei Methode GetTriggerValue.</p> <p>3.5.3.5 Auftragselement für Detektoren mit Zusatzinformation: Text in SetChannel (DigEingang) korrigiert.</p> <p>3.5.3.6 Auftragselement für erweiterte aggr. Det.-Werte: Text korrigiert.</p> <p>3.5.3.7 Auftragselement für Visualisierungsdaten: Methode 150 (GetTriggerValue) eingefügt.</p> <p>3.5.3.8.1 Signalbildcodierung präzisiert.</p> <p>3.5.3.9 Auftragselement DigAusgang: Format TriggerValue korrigiert, Text Verwendungszweck eingefügt.</p> <p>3.5.4 AP-Werte: RetCode bei Aufruf nicht existenter AP-Werte festgelegt.</p> <p>3.5.4.4 APWertGroup und APWertGroupRk: Text und Referenzen auf Subgruppen geändert. Beispiele eingefügt.</p> <p>3.5.5.4 Signalkammer: Hinweis eingefügt.</p> <p>3.5.6.1.1 Exclude Liste: Gelöscht 60307. (SupplyModification), Korrigiert: 60318 .</p> <p>3.5.6.2 Elementbeschreibung Betriebszustandsarchiv (gelöscht: Text „in OCIT“).</p> <p>3.5.6.3 Eigenschaften der Listen.</p>
V2.0 A02	PUBLIC	16.10.09	1.1 Version des Dokuments OCIT-I KD auf V1.0 korrigiert.

V2.0 A03	PUBLIC	18.06.10	<p>1.1 Version des Dokuments OCIT-I KD auf V1.1 korrigiert.</p> <p>2.3.4 Anforderungen an die Lichtsignalsteuergeräte: Hinweis auf das Dokument OCIT-O_Protokoll_V2.0 zur Größe der TCP-Telegramme eingefügt.</p> <p>2.10 Vorgangskennung: Hinweis eingefügt.</p> <p>3.1.2 Objekt SupplyTransaction: Methode 122 „ReadVDExt“ eingefügt.</p> <p>3.2 Versionierung: Hinweis in Tabelle 1: „Beschreibung der Versionierungsdaten“ entfernt.</p> <p>3.3.3.2.2 Tagesplan: Text in Zeile Uhrzeit präzisiert.</p> <p>3.3.2.3 Objekt SignalprogrammV: Text bei Eprogramm.Nr / Aprogramm.Nr geändert (falls definiert entfernt) und Hinweis auf Prüfung durch OCIT-I Versorgungsdatenserver eingefügt.</p> <p>3.4.1 OCIT konforme Konstellationen der zentralen Schaltwünsche: Text über der 2. Tabelle (Kombinationen TKZustand...) entfernt. Text in der 2. Tabelle in Zeilen 010 und 011 geändert.</p> <p>3.4.10 Objekt ZentralenSchaltwunsch: In Spalte "Modifikationen[0 15]" letzte Zeile neu.</p> <p>3.4.18 Objekt IBetriebsart: Hinweis ergänzt.</p> <p>3.5.2.2 Auftrag bei Abtaständerung: Text entfernt: Auftragsselement darf keine zusammengesetzte Struktur sein.</p> <p>3.5.2.3 Auftrag Abtaständerungen mit Wertevergleich: Text entfernt: Auftragsselement darf keine zusammengesetzte Struktur sein.</p> <p>3.5.3.1 bis 3.5.3.9: Wert TriggerValue präzisiert.</p> <p>3.5.3.10 Kombinationen von Aufträgen und Auftragsselementen: Neues Kapitel mit Tabelle.</p> <p>3.5.6.3 Eigenschaften der Listen: Hinweise geändert und ergänzt.</p>
V2.0 A04	PUBLIC	18.06.12	<p>1.2 Hinweis zu OCIT-C eingefügt.</p> <p>2.7 Relative Knoten: Regel für Nummerierung eingefügt.</p> <p>2.9.1 Zeitschaltwerte und Umlauf: Definition präzisiert, Hinweis eingefügt.</p> <p>3.1.1 Objekt Transaction: Methode 101, RetCode präzisiert.</p> <p>3.1.2 Objekt Supply Transaction: Methode 120 InitSupplyTransaction, Blocks: VDArt[0..3] und Methode Get Blocks präzisiert.</p> <p>3.3.3.2.1 Projektspezifische Modifikationen der Schaltuhr: Nummerierung präzisiert.</p> <p>3.2.2.3 LsaVersion: Hinweis eingefügt.</p> <p>3.2.2.4 LsaVersionPlus eingefügt.</p> <p>3.3.3.2.1 Neues Kapitel: Projektspezifischen Modifikationen der Schaltuhr</p> <p>3.3.2.4 Versatzzeitenmatrix: Hinweis zur Verwendung eingefügt.</p> <p>3.4 Zentrale Schaltwünsche: Text Teilknoten präzisiert.</p>

			<p>3.4.1 „OCIT-O konforme Konstellationen der zentralen Schaltwünsche“ und 3.4.6 „Objekt ZTeilKnoten“: TKZustand Ein = 1 definiert.</p> <p>3.4.8.1, 3.4.8.3, 3.4.8.4, 3.4.8.4.1: Methode Get eingefügt.</p> <p>3.4.11 bis 3.4.18: Hinweise zur lokalen Störabschaltung eingefügt.</p> <p>3.5.2.2 Auftrag bei Abtaständerungen: Hinweis zum Intervall 0 eingefügt.</p> <p>3.5.3.10 Kombinationen von Aufträgen und Auftragsselementen: Kombination von AuftragZykl 1:403 und AEBinaer 1:431 als nicht sinnvoll gekennzeichnet.</p> <p>3.5.4 AP-Werte: Hinweis auf Objekttyp von VA-Parametern eingefügt.</p> <p>3.5.4.1 APWert (1:505): Beschreibung präzisiert.</p> <p>3.5.4.1 Typen der AP-Werte korrigiert: ULONG zu LONG.</p> <p>3.5.4.1.2 APWertLong und 3.5.4.2.2 APWertRkLong korrigiert (LONG anstelle von ULONG).</p> <p>3.5.6 Archive: Definition der Einträge ins Betriebszustandsarchiv präzisiert.</p> <p>Glossar: Hinweis zu IPv4 eingefügt.</p>
V2.0 A04	Public	19.04.13	<p>Textänderung: Hauptmeldung geändert in Meldung, Nebmeldung geändert in Zusatzmeldung.</p> <p>3.3.3.2.6 Objekt Zeitbereich (1:664): Bei den jährlich wiederkehrenden Zeitbereichen wird das Startjahr und das Endjahr auf den NULLVALUE gesetzt. Der jährlich wiederkehrende Zeitbereich kann auch über den Jahreswechsel hinausgehen .</p>
V2.0 A04	Public	23.10.13	<p>3.3.2.3 Objekt SignalprogrammV: Der Eintrag in VTMinFreigabe.Nr und VTMinGesperrt.Nr bei Verwendung der Sicherheitszeiten wurde hinzugefügt.</p>

Spezifikationen

Das **OCIT-Outstations Konfigurationsdokument OCIT-O KD Vx.x** enthält eine Übersicht über alle von der ODG urheberrechtlich verwalteten Spezifikationen und ordnet Versionen und Ausgabestände nach:

- zusammengehörenden Spezifikationen der Schnittstelle „OCIT-Outstations für Lichtsignalsteuergeräte“ mit Referenz auf die dazugehörigen OCIT-Instations und OCIT-C Spezifikationen (siehe dazu Hinweis in 1.2),
- gibt Hinweise zum Einsatz der Übertragungsprofile und
- enthält eine Übersicht über Pakete von Spezifikationen für Schnittstellen, für deren Nutzung von der ODG eine Schutzgebühr verlangt wird

Der jeweils aktuelle Stand ist auf www.ocit.org veröffentlicht.

1 Einführung

In diesem Dokument werden alle für die OCIT-Schnittstelle zwischen einer Zentrale und den Lichtsignalsteuergeräten relevanten Funktionen festgelegt. Die herausragend neue Funktion in OCIT-O Lstg V2.0 ist die Möglichkeit der Standard-Datenfernversorgung der Lichtsignalsteuergeräte.

1.1 Unterstützte Funktionen

Die Schnittstelle OCIT-Outstations für Lichtsignalsteuergeräte in der vorliegenden Version basiert auf den vorne aufgeführten Referenzspezifikationen.

Eine OCIT-Outstations-Schnittstelle kann unterschiedliche Übertragungsprofile benutzen, die in den optionalen Definitionen festgelegt sind.

Es ist nicht verpflichtend, dass an OCIT-Outstations betriebene Gerätschaften alle in den Referenzspezifikationen festgelegten Funktionen unterstützen. Sie unterstützen nur diejenigen Funktionen, die für den jeweiligen Zweck und Ausbau notwendig sind. So wird z. B. ein Lichtsignalsteuergerät für Fußgängerüberwege weniger Funktionen unterstützen als ein Gerät mit verkehrsabhängiger ÖPNV-Bevorzugung. Die Nichtverfügbarkeit eines von der Zentrale aufgerufenen Leistungsmerkmals muss zu einer erkennbaren Reaktion (Returncode) des Lichtsignalsteuergeräts führen.

Die Spezifikationen der Schnittstelle OCIT-Outstations Version 2.0 für Lichtsignalsteuergeräte sind rückwärtskompatibel zu Zentralen mit OCIT-O Version 1.x.

Hinweis: In der Version 2.x wurden die zentralen Schaltkombinationen eindeutig definiert, in der Version 1.x war dies jedoch nicht der Fall. Wenn Lichtsignalsteuergeräte mit Version 2.x an Zentralen mit Version 1.x betrieben werden, kann daher bei Geräten unterschiedlicher

Hersteller weiterhin ein abweichendes Schaltverhalten auftreten. Abhilfe ist durch Anpassung der zentralen Schaltwünsche an die Festlegungen nach Version 2.x möglich.

Die Spezifikationen der Schnittstelle OCIT-Outstations Version 2.0 beinhalten Funktionen, deren Datenmodelle in OCIT-Instations spezifiziert sind und sind kompatibel mit den im OCIT-Instations Konfigurationsdokument OCIT-I KD Vx.x aufgeführten Spezifikationen zu

- „OCIT-Instations Versorgungsdaten Datenmodelle und Objekte Lichtsignalanlagen“ und
- „OCIT-Instations Prozessdaten Datenmodelle und Objekte Lichtsignalanlagen“.

Neue oder erweiterte Funktionen in OCIT-O Lstg V2.0:

2.10 Vorgangskennung erweitert um Versorgungs- und Prozessdatenserver

3.1 Übertragung von Versorgungsdaten

3.2 Versionierung der Versorgungsdaten

3.3 Versorgungsobjekte

3.4.10 ZentralenSchaltwunsch: Neue Methode „Schalte Knoten“

3.5.3.4 Auftragsselement für blockweises Lesen von AP-Werten

3.5.3.5 Auftragsselement für Detektoren mit Zusatzinformationen

3.5.3.6 Auftragsselement für erweiterte aggregierte Det.-Werte

3.5.3.7 Auftragsselement für Visualisierungsdaten

3.5.4.4 APWertGroup und APWertGroupRk

3.5.5 Detektoren und Signale

3.5.6 Archive der Lichtsignalsteuergeräte (dynamisches Archiv 31 für Prozessdaten)

Neue Funktionen in OCIT-O Lstg V2.0 Ausgabe 03:

3.1.2 Objekt SupplyTransaction: Methode 122 „ReadVDEExt“ eingefügt.

Neue Funktionen in OCIT-O Lstg V2.0 Ausgabe 04:

3.2.2.4 LSAVersionPlus. Verwendungshinweis in 3.2.2.3.

1.2 Unterstützung von OCIT-I Schnittstellen

Die Lichtsignalsteuergeräte mit OCIT-O Lstg Version 2 stellen standardisierte Daten und Funktionen bereit, die auf Festlegungen in den OCIT-I Spezifikationen für LSA beruhen.

Folgende OCIT-I Schnittstellen werden unterstützt:

- **OCIT-I Versorgungsdaten (OCIT-I VD)**

Das herausragende Merkmal von OCIT-O Lstg V2.0 ist die standardisierte Fernversorgung der Lichtsignalsteuergeräte von einem Planungsplatz aus. Dazu wurden Versorgungsdaten, die aus verkehrstechnischen Gründen häufig geändert werden müssen, in OCIT-O standardisiert. Diese in OCIT-O standardisierten Versorgungsdaten werden als "Anwenderversorgung" bezeichnet. Sie sind ein Teil der in OCIT-I VD festgelegten, wesentlich umfangreicheren Daten, die von den Planungsplätzen genutzt werden. Die Festlegungen dazu finden sich als Standard sowohl in der OCIT-I Dokumentation (OCIT-I VD-DM-LSA) als auch korrespondierend dazu in diesem Dokument. Die Formatumsetzung von OCIT-O auf OCIT-I übernimmt die OCIT-Instations Komponente OCIT-I VD Server. Die Realisierung des OCIT-VD Servers erfolgt gemäß den Festlegungen in OCIT-I VD-DM-LSA und OCIT-O Lstg V2.0 Spezifikation.

- **OCIT-I Prozessdatenerfassung (OCIT-I PD)**

Prozessdaten sind Daten und Messwerte, die vom Lichtsignalsteuergerät erfasst werden. Die Erfassung und Bereitstellung der Prozessdaten im Format OCIT-O wurde bereits mit der ersten OCIT-O Version realisiert. In OCIT-O Lstg V2.0 kommen die erweiterten Detektorwerte hinzu (siehe Pkt. 3.5.2.6). Weitere Festlegungen dazu werden in diesem Dokument nicht getroffen. Die Formatumsetzung von OCIT-O auf OCIT-I übernimmt die OCIT-Instations Komponente OCIT-I PD Server. Die Realisierung des OCIT-PD Servers erfolgt gemäß den Festlegungen in OCIT-I PD-DM-LSA und OCIT-O Lstg V2.0 Spezifikation.

Hinweis: Die Funktionen von OCIT-I VD-DM-LSA und OCIT-I PD-DM-LSA können seit November 2011 auch durch OCIT-C Funktionen abgedeckt werden. Die entsprechenden Schemata sind auf <http://www.ocit.org/downloadOCIT-C.htm> veröffentlicht. Formatumsetzungen erfolgen wie bei OCIT-I über entsprechende Server. Die OCIT-C Version 1 entspricht der in Kürze erscheinenden Norm DIN V VDE V 0832 - Straßenverkehrs-Signalanlagen - Teil 601 und Teil 602: Schnittstelle zwischen zentralen Einrichtungen zum Austausch verkehrsbezogener Daten.

2 Geräte- und Systemfunktionen

In diesem Kapitel finden sich Festlegungen, die für den Betrieb der definitionsgemäßen Lichtsignalsteuergeräte notwendig sind und die keine Festlegung als OCIT-Outstations-Objekte verlangen.

2.1 Lichtsignalsteuergeräte mit OCIT-O Schnittstelle

Auf Grund des Zeitverhaltens des OCIT-Outstations-Protokolls sind OCIT-Lichtsignalsteuergeräte speziell für Einsatz in dezentral aufgebauten Systemen gebaut. Sie beherrschen komplexe lokale Verkehrsabhängigkeiten und können Verkehrsmesswerte erfassen und verarbeiten („intelligente Steuergeräte“). Sie besitzen folgende charakteristischen Eigenschaften:

- Sie verfügen über leistungsfähige Mikroprozessoren, die komplexe Verkehrsabhängigkeiten lokal beherrschen und eine Verarbeitung von Messwerten durchführen.
- Sie verfügen über genaue Uhren, die Synchronisiervorgänge steuern, und deren Zeit zur Kennzeichnung von Ereignissen dient.
- Schaltvorgänge werden durch Signalprogramme gesteuert, wobei folgende Vorgaben gemacht werden:
 - Vordefinierte Signalpläne, die entweder im Gerät gespeichert sind, und/oder über die Zentrale im laufenden Betrieb versorgt werden können, werden über Schaltbefehle der Zentrale oder über interne Schalttabellen ausgewählt.
 - Wird eine lokale verkehrsabhängige Logik verwendet, werden ausgewählte Signalprogramme entsprechend der Verkehrssituation variiert.

- Die verkehrsabhängige Logik ihrerseits ist durch Parameter auf verschiedene Situationen einzustellen.
- Nicht für den Betrieb an der Schnittstelle OCIT-O Lstg ausgelegt sind folgende Lichtsignalsteuergeräte:
 - So genannte Schaltgeräte, deren Kennzeichen es ist, dass Schaltvorgänge vorzugsweise von der Zentrale veranlasst werden und innerhalb von einer Sekunde ausgeführt werden und die bei Ausfall der Zentrale nur einen Notbetrieb gewährleisten.
 - Gruppensteuergeräte, daher Geräte, die zentrale Steueraufgaben übernehmen, und kleinere Gerätegruppen über eigene Schnittstellen steuern.

2.2 Fernversorgung von Lichtsignalsteuergeräten

In OCIT-I werden die von Planungsplätzen erzeugten Versorgungsdaten für Lichtsignalsteuergeräte spezifiziert. Das Planungsergebnis für eine Geräteversorgung ist eine OCIT-I VD-DM-LSA konforme XML-Datei (in diesem Papier kurz als XML-Versorgungsdatei bezeichnet), die die in Blöcke gegliederten Versorgungs- und Planungsdaten enthält.

Eine standardisierte Untermenge der Versorgungsdaten (die sog. Anwenderversorgung) kann von der Zentrale oder vom VIAP aus, in standardisierter Weise in Lichtsignalsteuergeräte die die Festlegungen in OCIT-O Lstg V2.0 unterstützen, übertragen werden. Diese Anwenderversorgung ist in jedem störungsfreien Gerätezustand möglich. Die nicht standardisierten Versorgungsteile werden mit proprietären Mitteln versorgt.

2.3 Versorgungswege

Das als XML-Versorgungsdatei vorliegende Planungsergebnis (als Gesamtversorgung) kann nicht ohne weiteres in das Lichtsignalsteuergerät übertragen werden:

- Das Planungsergebnis für eine Geräteversorgung ist eine OCIT-I VD-DM-LSA konforme XML-Datei, die die gesamten in OCIT-I standardisierten Versorgungsdaten enthalten kann. Lichtsignalsteuergeräte mit OCIT-O Lstg V2.0 unterstützen davon das Subset „Anwenderversorgung“, das sind verkehrstechnisch häufig zu ändernden Versorgungsdaten. Die Anwenderversorgung ist in OCIT-I und OCIT-O korrespondierend spezifiziert.
- Die in der XML-Versorgungsdatei enthaltenen Daten der OCIT-I „Gerätetechnik“ und „Sicherheitstechnik“ können nur herstellereinspezifisch und nicht über den OCIT-I VD Server versorgt werden.
- In den Spezifikationen zu OCIT-Instations werden andere Datenformate und Protokolle als in OCIT-Outstations verwendet. Es ist daher notwendig einen Umsetzer, den so genannten OCIT-I VD Server in die Versorgungskette einzubinden. Dieser extrahiert die in OCIT-I enthaltenen und von OCIT-O Lstg V2.0 unterstützten Versorgungsdaten und setzt sie in das Format OCIT-O Lstg um. Die Realisierung des

OCIT-VD Servers erfolgt gemäß den Festlegungen in OCIT-I VD-DM-LSA (siehe Referenzspezifikationen) und OCIT-O Lstg V2.0.

Der OCIT-I VD Server kann integriert werden (Abbildung 1):

- in die zentrale Systemrealisierung
Die Versorgungsdaten gelangen hier vom Planungswerkzeug über die Schnittstelle OCIT-I VD zum OCIT-I VD-Server, werden dort in das OCIT-O Format gewandelt und über die zentrale Kommunikationskomponente zum Lichtsignalsteuergerät übertragen, und /oder
- in ein Planungs- oder Versorgungswerkzeug
Die Versorgungsdaten gelangen hier vom einem im Werkzeug integrierten OCIT-I VD-Server über einen Systemzugang mittels OCIT-O Lstg zum Lichtsignalsteuergerät.

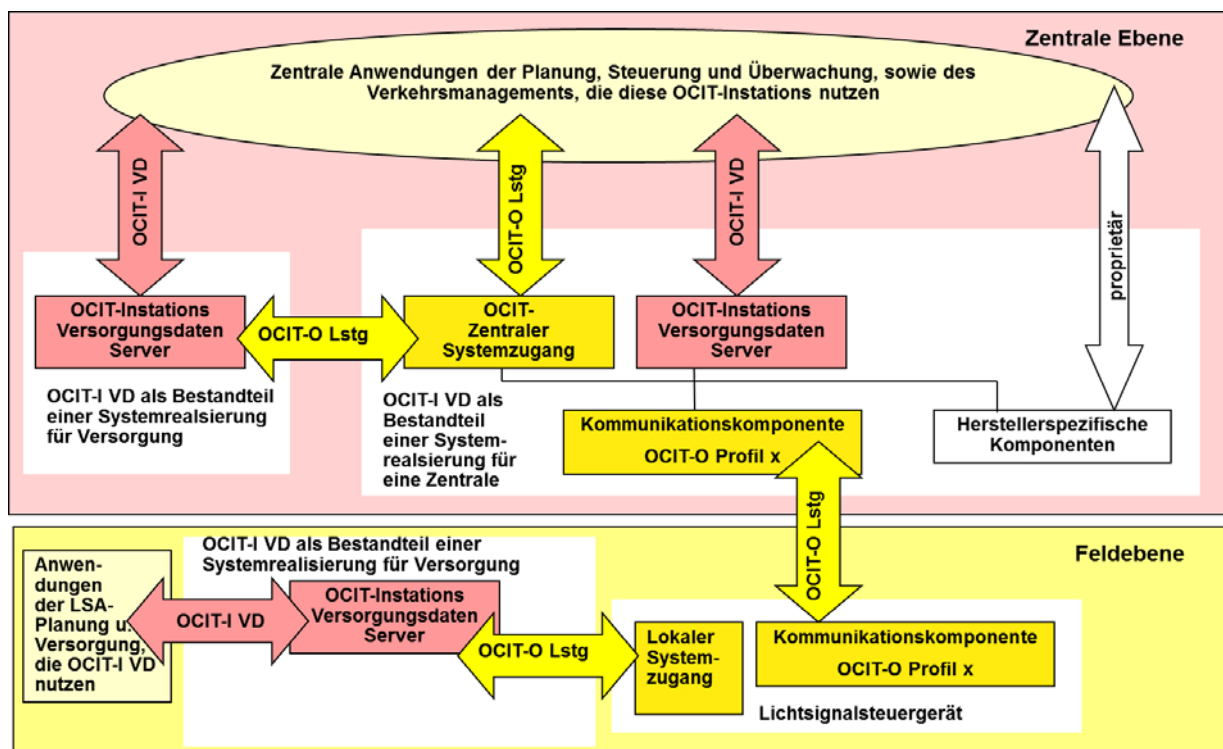


Abbildung 1: Anordnung des OCIT-I VD-Servers ¹

Der Versorgungsvorgang für die von OCIT-O Lstg V2.0 unterstützten Daten kann herstellerübergreifend direkt vom Standard-Planungs- und/oder Versorgungswerkzeug aus gestartet werden. Der OCIT-I VD Server steuert dann den geräteseitigen Versorgungsablauf.

¹ Der „OCIT-O Lokaler Systemzugang“ wird in OCIT-O Lstg V2.0 nicht angeboten.

Hinweis: Das Protokoll OCIT-I VD-SP in Version 1.0 sieht „blockierende Funktionsaufrufe“ vor. Diese Einschränkung kann dazu führen, dass beim Aufruf von Funktionen am VD-Server, so lange gewartet werden muss, bis das Ergebnis des Aufrufs gemeldet wird. Dabei können im Dialog zwischen VIAP und Lichtsignalsteuergeräten lange Antwortzeiten – im Bereich von mehreren Minuten – auftreten.

Die Antwortzeiten werden beeinflusst durch:

- Bearbeitungszeit im VD-Server
- Übertragungszeit zur LSA (Geschwindigkeit und Dateigröße)
- Zeit für die Überprüfung der Daten in den Lichtsignalsteuergeräten
- Je nach VIAP-Bedienkonzept Zeit für die Aktivierung (bis zu 2 Signalplanumläufe)
- Bei Profil 2 mit GSM ist wegen der gegenüber Festverbindungen geringeren Übertragungskapazität und der für den Verbindungsaufbau benötigten Zeit mit weiteren, im Dokument OCIT-O-Profil_2 im Anhang beschriebenen Einschränkungen zu rechnen.

2.3.1 Versorgungsdaten

Die Versorgung des Lichtsignalsteuergeräts teilt sich von der Handhabung her auf, in

- die in OCIT-I VD standardisierte Anwenderversorgung der Verkehrstechnik die von einem Standard-Versorgungswerkzeug aus erfolgen kann, und
- die Herstellerversorgung von in OCIT-I VD standardisierten Daten der Sicherheits- und Gerätetechnik, sowie von proprietären Daten. Die Herstellerversorgung wird mit Mitteln der jeweiligen Gerätehersteller vorgenommen.

Hinweis: Die Sorgfaltspflicht auf Stimmigkeit der Versorgung liegt beim Versorger. In den Geräten ist vor der Aktivierung der Versorgung nur eine grobe Prüfung möglich. Es ist daher nicht ausgeschlossen, dass das Gerät als Folge einer fehlerhaften Versorgung eine Sicherheitsabschaltung durchführt.

Die Versorgungsdaten sind entsprechend ihrer Aufgabe in Blöcke gegliedert. Entsprechend den Festlegungen erfolgt die Anwenderversorgung immer blockweise, das heißt, auch bei einer Änderung nur eines Wertes werden immer alle Daten eines Blocks versorgt.

Folgendes Schema zeigt die Versorgungsdatenblöcke:

In OCIT-I VD standardisierte Versorgungsdaten, herstellerübergreifend versorgbar und auslesbar Anwenderversorgung				In OCIT-I VD teilweise standardisierte Versorgungsdaten, nur proprietär versorgbar und auslesbar Herstellerversorgung			
Verkehrstechnik				Gerätetechnik		Sicherheitstechnik	
Verkehrstechnische Grunddaten / Festzeit	Daten mit Netzbezug	VA-Steuerverfahren	VA-Parameter	OCIT-I VD Versorgungsdaten	Proprietäre Daten	OCIT-I VD Sicherheitsdaten	Proprietäre Daten
Einschaltprogramme	Kopfdaten	Anwendungsspezifische Dateien (Binär)	Anwendungsspezifische Dateien (Binär)	Detektoren bzw. digitale Eingänge	Gerätetechnische Versorgungsdaten	Sicherheitsrelevante Unverträglichkeitsmatrix	Übergänge, Farbfeindlichkeiten..
Ausschaltprogramme	JAUT: Tagesplan			Signalgruppen bzw. digitale Ausgänge		Sicherheitsrelevante Zwischenzeitenmatrix	
Signalprogramme	JAUT: Wochenplan			Zuordnung zum Teilknoten		Sicherheitsrelevante Mindestfreigabezeiten	
Versatzzeitenmatrizen	JAUT: Sondertag jährlich			Übergangszeiten		Sicherheitsrelevante Mindestsperrzeiten	
VT-Zwischenzeitenmatrizen	JAUT: Sondertag Aufzählung			ÖV-Meldepunkte und Meldekettens			
VT-Mindestfreigabezeiten	JAUT: Zeitbereich						
VT-Mindestsperrzeiten							

Abbildung 2: Schema der Versorgungsdatenblöcke

Inhalt der Versorgungsdatenblöcke der **Anwenderversorgung**:

- **Block 1: Verkehrstechnische Grunddaten / Festzeit**
 - n Einschaltprogramme (EProgramm)
 - n Ausschaltprogramme (AProgramm)
 - n Signalprogramme (SignalprogrammV)
 - n Versatzzeitenmatrizen (Versatzzeitenmatrix)
 - n Verkehrstechnische Zwischenzeitenmatrizen (VTZwischenzeitenmatrix)
 - n Verkehrstechnische Mindestfreigabezeiten (VTMinFreigabe)
 - n Verkehrstechnische Mindestsperrzeiten (VTMinGesperrt)
- **Block 2: Daten mit Netzbezug**
 - Kopfdaten
 - Jahresautomatik / JAUT/ Schaltuhr: Tagesplan, Wochenplan, Sondertag Jaehrlich, Sondertag Aufzaehlung, Zeitbereich
- **Block 3: VA-Steuerverfahren**
 - n Anwendungsspezifische Dateien (Binaer VASteuerverfahren)
- **Block 4: VA-Parameter**
 - n Anwendungsspezifische Dateien (Binaer VAParameter)

2.3.2 Versionierung

Die Versionierung von Versorgungsständen hat den Zweck, jederzeit nachzuvollziehen zu können, dass Versorgungsänderungen passiert sind, und zwar unabhängig davon ob sie zentral oder lokal vorgenommen wurden. Das Versionierungsverfahren ist zwischen Lichtsignalsteuergerät, OCIT-I VD-Server und Versorgungswerkzeug durchgängig und umfasst sowohl die Anwender- als auch Teile der Herstellerversorgung. Spezifikation siehe Pkt.3.2.

2.3.3 Anforderungen an den Versorgungsdatenserver

- Der OCIT-I VD- Server muss sicherstellen, dass nur ein Versorgungsvorgang – der aus mehreren Blöcken bestehen kann – zu einer Zeit erfolgt und dass dieser Vorgang abgeschlossen wird.
- Der OCIT-I VD-Server überträgt immer alle Versorgungsdaten eines Blocks und die dazugehörigen Versionierungsdaten. Der funktionelle Inhalt der Versorgungsblöcke ist im Standard festgelegt und darf nicht verändert werden. Eine spätere Änderung der Blöcke hätte Auswirkungen auf die Software der Lichtsignalsteuergeräte. Durch die starren Inhalte ergeben sich Vorteile für die Nachvollziehbarkeit und Prüfbarkeit.
- Die Versorgungsdaten werden als XML und Trace-Datei im Dateisystem des OCIT-I VD Servers gespeichert (mandatory).

2.3.4 Anforderungen an die Lichtsignalsteuergeräte

- TCP-Telegrammgröße: siehe OCIT-O_Protokoll_V2.0 „Verwendung der OSI-Schicht 4 Protokolle (UDP, TCP)“

Hinweis: Bei sehr großen Versorgungen inkl. VA-Parametern kann es zu einer Überschreitung der max. möglichen Telegrammgröße kommen. Der VD-Server kann in solchen Fällen die Versorgung blockweise senden bzw. lesen. Sollte in einem Feldgerät durch ein Hersteller-Tool eine zu große Versorgung eingespielt worden sein, muss das FG bei einem BTPPL-Leseversuch mit dem RetCode TOO_MANY antworten.

Es ist beabsichtigt ab der nächsten Version eine Fragmentierung einzuführen, durch welche die Größenbeschränkung der Telegramme aufgehoben wird.

- Die stand. Versorgungsdaten der Anwenderversorgung werden als OCIT-O Objekte abgebildet.
- Es werden die bestehenden OCIT-O Methoden, Fehlermeldungen etc. verwendet. Versorgungsobjekte werden mit btppl als Nachricht mit niedriger Priorität übertragen.
- Bei einer Änderung der Versorgung eines Blocks, muss die gesamte, dem Block zugeordnete alte Versorgung gelöscht werden um nicht Reste davon weiter aktiv zu halten.
- Die standardisierten Daten der Anwenderversorgung können so wie sie im Lichtsignalsteuergerät wirksam versorgt sind ausgelesen werden.
- Falls das Lichtsignalsteuergerät bestimmte Werte der Versorgungsdaten nur in abweichender Form unterstützt, muss die Versorgung abgewiesen werden. Eine automatische Korrektur dieser Werte durch das Lichtsignalsteuergerät ist nicht zulässig.
- Die Versorgungsdaten der Herstellerversorgung werden in OCIT-O nicht spezifiziert und können daher nur mit herstellersizifischen Mitteln versorgt und gelesen werden. Mit den Mittel der Schnittstelle OCIT-O Lstg V2.0 sind jedoch über Instance-Info oder ExtendedInstanceInfo die Outstations-Nummern der Signalgruppen- und Detektoren auslesbar. Die Namen der Signalgruppen- und Detektoren findet man über die entsprechenden Objekte.

2.4 Übertragungsgeschwindigkeit

Mit einer Übertragungsgeschwindigkeit von 19200 bis 28800 bit/sec ist nach den bisherigen Einsatzerfahrungen die intensive Nutzung des OCIT-Signalisierungsarchivs (Darstellung der aktuellen Signalisierung in der Zentrale / Visualisierung) und anderer Archive möglich. Bei Übertragungsgeschwindigkeiten unter 9600 bit/s sind diese Möglichkeiten stark eingeschränkt. Sehr schlechte Übertragungswege erlauben oft nur mehr den Betrieb mit 2400 bit/s. Dies ist das absolute Minimum, mit dem nur noch Bedienen und Melden möglich ist.

2.5 Synchronisierung u. Rückrechenverfahren

Die Synchronisierung der Signalisierung von Lichtsignalsteuergeräten im Straßennetz erfolgt uhrengesteuert. Mit Hilfe von Rückrechenverfahren werden die Umlaufzähler der einzelnen Geräte auf einen bestimmten Bezugszeitpunkt synchronisiert. Ein Steuergerät gilt hierbei als synchron, wenn die Umlaufsekunde des Umlaufzählers TX identisch mit dem „modulo TU-Wert“ der seit dem jeweiligen Bezugszeitpunkt abgelaufenen Sekunden ist. Als TU (Umlaufzeit eines Signalplans) wird die Umlaufzeit des aktuell laufenden Signalplans herangezogen. Bei Signalplanwechsel wechselt der TU-Wert in Abhängigkeit vom verwendeten Umschaltverfahren mit dem Vollzug des Signalplanwechsel. Sinnvollerweise wird die Synchronisierung bei Anstoß eines Signalplanwechsels ausgesetzt und erst nach vollzogenem Wechsel im neuen Signalplan fortgesetzt.

Synchronitätsbedingung: $TX = (RRS + \text{SignalzeitenVersatz}) \% TU$

RSS: Rückrechensekunde, d. h. die aktuell abgelaufenen Sekunden seit Bezugszeitpunkt

TU: Umlaufzeit des aktuellen Signalplans

SignalzeitenVersatz: siehe Pkt. 3.3.2.3

Für das Rückrechnen haben sich vier verschiedene Bezugszeitpunkte und davon abgeleitet vier verschiedene Rückrechenverfahren eingebürgert. Das Rückrechenverfahren (RVV) ist daher projektspezifisch festzulegen, da es im System (Bestand + OCIT-O) gleich sein muss. Lichtsignalsteuergeräte mit OCIT-O Schnittstelle müssen mindesten folgende Rückrechenverfahren beherrschen:

- 1. RVV UTC (Bezugszeitpunkt 1.1.1970 0:00:00 Uhr, Universal Time Coordinated)**
Bei diesem einfachen aber wenig verbreiteten Rückrechenverfahren wird die aktuelle UTC – Zeit als Bezugszeitpunkt genommen. Für die Ermittlung der aktuellen Referenzsekunde muss die aktuelle UTC – Zeitsekunde modulo der Umlaufzeit des aktuell laufenden Signalplans genommen werden.
- 2. RVV 1.1 (Bezugszeitpunkt 1.1. 0:00:00 Uhr Lokalzeit aktuelles Jahr)**
Dieses weit verbreitete Rückrechenverfahren benutzt den 1.1. 0:00:00 Uhr Lokalzeit des aktuellen Jahres als Bezugszeitpunkt. Die Rückrechensekunde (RRS) ergibt sich aus den seit dem Bezugszeitpunkt gemäß der aktuellen lokalen Uhrzeit abgelaufenen Sekunden. Der Sommerzeitsprung von einer Stunde wird bei diesem Verfahren als abgelaufene Sekunden berücksichtigt, d. h. die RRS springt beim Sommerzeitwechsel um 3600 Sekunden, so als wäre die übersprungene Zeit tatsächlich abgelaufen. Beim Rücksprung gilt dies analog.
- 3. RVV 1.1.1980 (Bezugszeitpunkt 1.1.1980 0:00:00 Uhr Lokalzeit)**
Dieses weniger weit verbreitete Rückrechenverfahren benutzt den 1.1.1980 0:00:00 Uhr als Bezugszeitpunkt. Die Rückrechensekunde (RRS) ergibt sich aus den seit dem Bezugszeitpunkt tatsächlich abgelaufenen Sekunden. Der Sommerzeitsprung von einer Stunde wird bei diesem Verfahren nicht berücksichtigt, d. h. die RRS springt beim Sommerzeitwechsel nicht, sondern läuft kontinuierlich weiter.

4. **RRV Mitternacht (Bezugszeitpunkt 0:00:00 Uhr Lokalzeit des aktuellen Tages)**
 Dieses weniger weit verbreitete Rückrechenverfahren benutzt Mitternacht (0:00:00 Uhr) als Bezugszeitpunkt. Die Rückrechensekunde (RRS) ergibt sich aus den seit dem Bezugszeitpunkt gemäß der aktuellen Uhrzeit abgelaufenen Sekunden. Der Sommerzeitsprung von einer Stunde wird bei diesem Verfahren als abgelaufene Sekunden berücksichtigt, d.h. die RRS springt beim Sommerzeitwechsel um 3600 Sekunden, so als wäre die übersprungene Zeit tatsächlich abgelaufen. Beim Rücksprung gilt dies analog. Dies ist jedoch nur an den beiden Wechseltagen relevant und hat an allen anderen Tagen keine Auswirkung

2.5.1 Berechnungsbeispiele für Rückrechenverfahren

Sekundenwerte für die Rückrechenverfahren:

UTC_1_1_1980OFFSET = 315529200 (Offset der UTC-Zeit vom 1.1.1970 auf den 1.1.1980)

JAHRESSEKUNDEN = 31536000

TAGESSEKUNDEN = 86400

STUNDENSEKUNDEN = 3600

MINUTENSEKUNDEN = 60

RRV UTC:

20.3.07 16:30:00 Uhr	-> RRS: 1174404600	RefZeit bei TU = 70: 40
25.3.07 03:10:00 Uhr	-> RRS: 1174785000	RefZeit bei TU = 70: 60
20.4.07 16:50:22 Uhr	-> RRS: 1177080622	RefZeit bei TU = 70: 32

Berechnungsvorschrift: RefZeit = akt. UTC-Zeit % TU

RRV 1.1.:

20.3.07 16:30:00 Uhr	-> RRS: 6798600	RefZeit bei TU = 70: 60
25.3.07 03:10:00 Uhr	-> RRS: 7182600	RefZeit bei TU = 70: 40
20.4.07 16:50:22 Uhr	-> RRS: 9478222	RefZeit bei TU = 70: 12

Berechnungsvorschrift: RefZeit = ((Tag des Jahres * TAGESSEKUNDEN) + (akt. Stunde * STUNDENSEKUNDEN) + (akt. Minute * MINUTENSEKUNDEN) + akt. Sekunde) % TU

RRV 1.1.1980:

20.3.07 16:30:00 Uhr	-> RRS: 858875400	RefZeit bei TU = 70: 40
25.3.07 03:10:00 Uhr	-> RRS: 859255800	RefZeit bei TU = 70: 60
20.4.07 16:50:22 Uhr	-> RRS: 861551422	RefZeit bei TU = 70: 32

Berechnungsvorschrift: RefZeit = (akt. UTC-Zeit – UTC_1_1_1980OFFSET) % TU

RRV Mitternacht:

20.3.07 16:30:00 Uhr -> RRS: 59400 RefZeit bei TU = 70: 40

25.3.07 03:10:00 Uhr -> RRS: 11400 RefZeit bei TU = 70: 60

20.4.07 16:50:22 Uhr -> RRS: 60622 RefZeit bei TU = 70: 2

Berechnungsvorschrift: $\text{RefZeit} = (\text{akt. Stunde} * \text{STUNDENSEKUNDEN}) + (\text{akt. Minute} * \text{MINUTENSEKUNDEN}) + \text{akt. Sekunde}) \% \text{ TU}$

2.6 Teilknoten

Teilknoten sind zu einzelnen Signalisierungsbereichen zusammengefasste Signalgruppen eines Gesamtknotens (relativer Knoten), die zueinander nicht feindlich sind. Alle Teilknoten arbeiten zu einer bestimmten Zeit mit dem selben Signalprogramm. Teilknoten können von der Zentrale ein- und ausgeschaltet werden.

Ein OCIT-O Lichtsignalsteuergerät ab Version 2.0 besitzt immer mindesten einen Teilknoten. Der Gesamtknoten (oder auch relative Knoten) kann aus 1 bis max. 4 Teilknoten bestehen. Die Nummerierung beginnt bei 0.

Meldung der Teilknoten im IstVektor:

	Anzahl der gemeldeten Teilknoten			
Gesamtknoten (Rel. Knoten)	1 TK	2 TK	3 TK	4 TK
Index / Nummer im IstVektor	0	0 1	0 1 2	0 1 2 3

In der Nummerierung der Teilknoten darf keine Lücke sein.

Beim Schalten eines nicht vorhandenen Teilknotens liefert das Lichtsignalsteuergerät einen Fehler als Returncode.

Im Auslieferungszustand oder unbestimmten Zuständen folgen die Lichtsignalsteuergeräte dem Zustand des Gesamtknotens (TKZustand = 1).

2.7 Relative Knoten (mehrere Knoten innerhalb eines Lichtsignalsteuergerätes)

Das Adressierungsschema von OCIT-Outstations sieht vor, dass mit einem Gerät (theoretisch) bis zu 255 logisch voneinander unabhängige Knotenpunkte (relative Knoten) realisiert werden können. Die Nummerierung der Relative Knoten beginnt bei 0. Der Default-Wert bei einem Relativen Knoten ist 0. Jeder relative Knoten kann Teilknoten enthalten.

Hinweis: Nicht alle Hersteller können derartige (aufwändige) Geräte anbieten.

2.8 Zeitstempel

Meldungen vom Lichtsignalsteuergerät zur Zentrale, Archiv- oder Listeneinträge sind mit einem Zeitstempel mit einer Auflösung von einer Sekunde zu versehen. Der Zeitstempel gibt an, wann ein Ereignis aufgetreten ist. Format: UTC (siehe OCIT-O Protokoll).

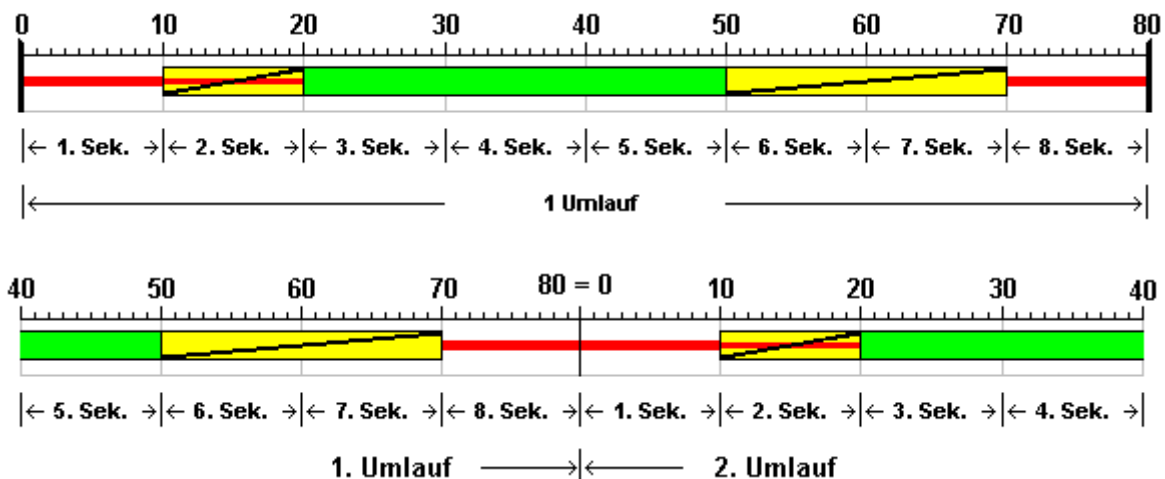
2.9 Zeitzählung

Die Zählung von Zeiten, z. B. der Umlaufzeit eines Signalplans, beginnt mit Sekunde 0. Sekunde 0 bezeichnet den Zeitabschnitt von Beginn bis zum Ende der ersten Sekunde.

Standardkonform sind Schaltzeitpunkte 0 bis TU-1, die Referenzzeit zeigt ebenfalls von 0 bis TU-1. Beispiel: Bei einem 60 Sekunden Umlauf muss das TX=0 auf den Beginn der vollen Minute liegen.

2.9.1 Zeitschaltwerte und Umlauf

- Der Start eines Umlaufs erfolgt immer bei TX=0.
- Der Umlaufzähler TX beginnt beim Schaltwert 0 (= Beginn der 1. Sekunde des Umlaufs) und endet beim letzten Schaltwert des Umlaufs ($< TU$).
- Die in den Versorgungsdaten angegebenen Zeiten haben eine Auflösung von 0,1 Sekunden.
- Die 1. Sekunde umfasst die Schaltwerte 0 bis 9.
- Wert 0 und Wert TU bezeichnen demnach einen identischen Zeitpunkt. Zugelassen ist der Schaltwert 0, nicht zugelassen ist der Schaltwert TX=TU.



Hinweis: Darstellung in 0.1 Sekunden-Auflösung!

Abbildung 3: Beispiel eines Programms mit einer Umlaufzeit TU = 8 Sekunden

Im Beispiel (Abbildung 3) bewirkt ein Zeitschaltwert 10 den Beginn einer Aktion zum Beginn der 2. Sekunde des Umlaufs.

Versorgungen, die Zeitschaltwerte enthalten, die ein Lichtsignalsteuergerät nicht unterstützt (z.B. Schalten in 0,1 Sekunden- Auflösung) werden bei der Aktivierung abgewiesen. Eine automatische Korrektur der zu versorgenden Zeitschaltwerte durch das Lichtsignalsteuergerät ist nicht zulässig!

Hinweis: Standardisiert ist das Verhalten auf der Schnittstelle. OCIT-O macht keine Aussage zur Visualisierung!

2.10 Vorgangskennung

Hinweis: Funktionsumfang wurde gegenüber der Basisfunktionalität und der Vorgängerversion erweitert um die Verursacher Versorgungs- und Prozessdatenserver. Versionsstand des Lichtsignalsteuergeräts beachten!

SYSJOBID (siehe Dokument OCIT-O Basis) dient zur Zuordnung von Meldungen zu Bedienvorgängen.

Verbindlichkeit: Die Vorgangskennung ist zwingend zu verwenden.

Beispiele für Objekte, die die Vorgangskennung mitführen: Zentrale Schaltwünsche, Ist-Zustandsmeldungen, Remote-Service, Versorgungstransaktionen, Meldungen. Details finden sich in den jeweiligen Objektdefinitionen.

Regel: Eingetragen wird die Vorgangskennung des Systemteils,

welcher die Kontrolle über einen Schaltzustand ausübt, auch wenn sich der Schaltzustand bei einem Wechsel des Auftrags nicht ändert (Beispiel: Wahl des Signalprogramms),

oder des Systemteils, der einen Vorgang ausgelöst hat (Beispiel: Versorgungstransaktion).

Im Folgenden die Spezialisierung für OCIT-O Lstg:

Vorgangskennung				Bezeichnung	
Herkunft			Auftragsnummer		
Systembezeichnung	Verursacher				
Teilsystem	Typ	Untertyp	Instanz		
1 = Zentrale 2 = Systemzugang 3 = Feldgerät	0 = keine Detaillierung		0 bis 63 0 bis 63 0 bis 65535	0 bis 65535 0 bis 65535 0 bis 63	
	1 = ZAUT	0 = keine Detaillierung 1 = Tagesplan 2 = Stadtfahrplan 3 -15 frei			Zeitautomatik / Schaltuhr (Stadtfahrpläne, z.B. für Fußballspiel)

Vorgangskennung					Bezeichnung
Herkunft				Auftragsnummer	
Systembezeichnung	Verursacher				
Teilsystem	Typ	Untertyp	Instanz		
	2 = VA-Logik	0 = keine Detaillierung 1 - 15 frei			verkehrsabhängige Logiken
	3 = Bedienzugänge	0 = keine Detaillierung 1 = Bediengerät integriert 2 = Bediengerät abgesetzt 3 = Service PC 4 = Sondereingriff 5 = Blockierung 6 - 15 frei			Bedienzugänge über lokale und zentrale Bediengeräte / PCs
	4 = Übertragungssysteme	0 = keine Detaillierung 1= ausgelaufener Schaltwunsch 2= Übertragungsstörung			Störungen des Übertragungssystems
	5 = Überwachungen	0 = keine Detaillierung 1 = Signalsicherung 2 = Zwischenzeitüberwachung 3 = Feindlichkeitsüberwachung 4 = Mindestgrünüberwachung 5 = Mindestrotüberwachung 6 = Umlaufüberwachung 7 -15 frei			Zustandsänderungen aufgrund interner Überwachungen
	6 = Versorgungsdatenserver	0 – 15 frei			
	7 = Prozessdaten-server	0 – 15 frei			
	8 - 15 frei				

Z. B: Bedienungsvorgang von zentraler Zeitautomatik (ZAUT), Zentrale 0:

1	1	1	0	Auftragsnummer	Tagesplan
---	---	---	---	----------------	-----------

Bedienung über lokales, abgesetztes Bediengerät (z.B. Handpanel), Lichtsignalsteuergerät 317:

3	3	2	317	Auftragsnummer	Abgesetztes Bediengerät
---	---	---	-----	----------------	-------------------------

Hinweis: „frei“ bedeutet: Freizuhalten für spätere Erweiterungen des Standards und nicht zur Verwendung für projektspezifische Lösungen!

Beispiel Lampenwechsel:

Der Servicetechniker wechselt Lampen am Lichtsignalsteuergerät Nummer 32. Dazu schaltet er mit seinem Service-PC vor Ort die Anlage zunächst aus, dann wechselt er die Lampen, führt einen Testlauf mit Signalprogramm 1 durch und meldet sich schließlich wieder ab:

Teilsystem=Lichtsignalsteuergerät(3)

Typ=Bedienzugang(3)

Untertyp=ServicePC(3)

Instanz=FNr(32)

Auftragsnummer(1)

==> SYSJOBID=11 0011 0011 0000000000100000 000001B=0xCCC00801

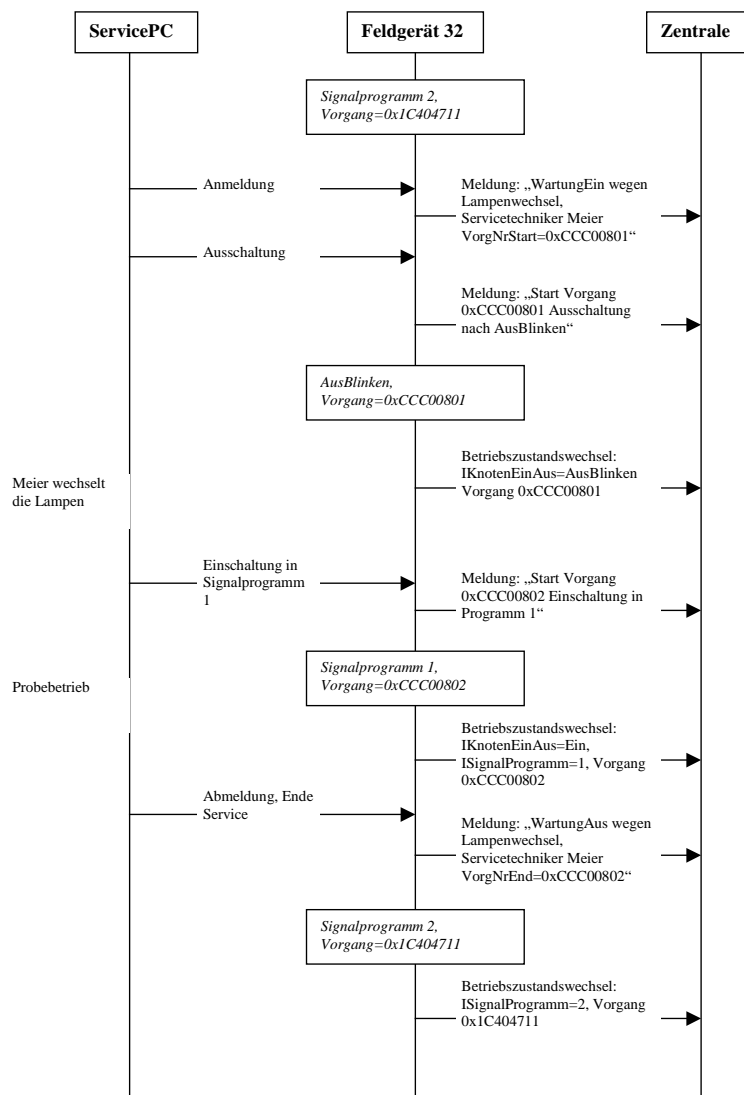


Abbildung 4: Ablaufschema zum Beispiel Lampenwechsel

3 Objektdefinitionen

In der zu diesem Dokument gehörenden Datei OCIT-O-Lstg-TYPE_V2.0.xml finden sich detaillierte Beschreibungen der Daten und Methoden der Objekte. Details, die aus Gründen der Übersichtlichkeit bei den folgenden Objektbeschreibungen nicht aufgeführt wurden, sind nur dort zu finden.

Hinweis: Es gilt die Festlegung zu den nummerierten Elementen aus OCIT-O Basis: Die Adressierung nummerierter Elemente wie Signalgruppen und Detektoren etc. beginnt mit dem Indexwert 1. Der Index wird nicht gemappt: Index 1 adressiert Element 1 usw. Damit ist sichergestellt, dass der Indexwert mit der von den Anwendern verwendeten Nummer eines nummerierten Elements übereinstimmt.

3.1 Übertragung von Versorgungsdaten

Versorgungsvorgänge werden durch Transaktionen geklammert. Eine Transaktion ermöglicht es den OCIT-I VD-Server zu versorgende Daten schrittweise ins Lichtsignalsteuergerät zu übertragen, zu überprüfen ob die übertragenen Daten konsistent sind diese dann zu einem definierten Zeitpunkt zu aktivieren.

Die Transaktion wird in zwei verschiedenen Zusammenhängen genutzt:

1. Es gibt die Möglichkeit Daten der Anwenderversorgung zu aktivieren. Dies ist umgesetzt durch das Objekt Versorgungstransaktion. Dieses Objekt stellt Funktionalitäten bereit, die neben der Grundfunktionalität der Transaktion den Versorgungsvorgang durch geeignete Meldungen dokumentieren.
2. Mit dem Objekt TransferParameterBlock können Änderungen von AP-Werten durch eine Transaktion geklammert werden.

Die Gemeinsamkeiten dieser beiden Arten von Transaktionen sind im Objekt Transaction zusammengefasst.

3.1.1 Objekt Transaction

Das **Objekt Transaction** ist die Basedomain für

- die Transaktion für die Übertragung der Daten der Anwenderversorgung (Supply-Transaction, Pkt. 3.1.2) und
- die Transaktion für das blockweise Schreiben von AP-Werten (TransferParameter-Block, Pkt. 0).

Die Übertragung von Daten für die Anwenderversorgung oder für Blöcke von AP-Werten wird vom VIAP oder der Zentrale aus initiiert.

Grundsätzlich benötigt das Lichtsignalsteuergerät für die zu versorgenden Daten Zwischenspeicher (Datenpuffer) und zwar getrennt für die

- Versorgungstransaktion (Versorgungsdatenpuffer) und
- die AP-Wert Blöcke (AP-Blockpuffer).

Zu einem Zeitpunkt kann immer **nur je eine** Versorgungstransaktion und **eine** Übertragung von AP-Werten laufen.

Im den Datenpuffern werden die Daten der Transaktionen zwischengespeichert, wobei sich jeweils nur ein Stand dieser Daten im den entsprechenden Puffern befinden kann. Die Daten im Puffer können bereits aktiviert (d.h. von Anwendungen im Lichtsignalsteuergerät übernommen) sein oder auf ihre Aktivierung warten. Der Puffer kann gelöscht oder mit neuen Daten gefüllt werden. Festlegungen zur Größe siehe Punkte 3.1.2 und 0.

Zu Beginn einer Transaktion muss initialisiert werden. Dadurch werden alle bereits im Versorgungsdatenpuffer des Lichtsignalsteuergerät enthaltenen Daten gelöscht. Danach wird der Versorgungsdatenpuffer im Lichtsignalsteuergerät mit den zu versorgenden Daten mit der Methode AddChangeSet gefüllt. Das Lichtsignalsteuergerät prüft, ob die im Puffer angelangten Versorgungsdaten von den Anwendungen im Lichtsignalsteuergerät angenommen werden können.

Ist die Übertragung vollständig erfolgt und sind alle Daten geprüft, so werden die Änderungen mit der Methode Activate in die entsprechenden Anwendungen des Lichtsignalsteuergerät übernommen. Eine eventuell laufende Transaktion kann durch die Methode Abort abgebrochen werden.

Die Transaktion kann sich in einem der folgenden Zustände befinden:

none	Die Transaktion ist nicht initialisiert.
empty	Die Transaktion ist initialisiert und leer.
receiving	Die Transaktion enthält bereits Änderungen, ist jedoch noch nicht geprüft und nicht aktiviert.
checkFailed	Die Änderungen sind geprüft und nicht aktivierbar.
checked	Die Änderungen sind geprüft und als aktivierbar eingestuft worden.
complete	Die Änderungen sind komplett übertragen. Sie sind jetzt nicht mehr änderbar, jedoch aktivierbar.
activationSet	Ein Aktivierungstermin wurde gesetzt.
activating	Die Anwenderversorgung wird im Lichtsignalsteuergeräte aktiviert (Dauer voraussichtlich 1 bis 2 Umläufe).

Es ist nicht garantiert, dass die Daten der Transaktion vor ihrer Aktivierung persistent gehalten werden können und z.B. einen Netzausfall überstehen. In diesem Fall wäre die Transaktion nach der Störung wieder im Zustand „none“. Dieser Zustandsübergang wird wie unten beschrieben durch eine Meldung „VersorgungEnde“ mit dem Zusatzmeldungsteil „TransactionFailed“ dokumentiert.

Transaction (1:710)

Transaction		
METHOD	Name	Beschreibung
100	Init	Die Transaktion wird zurückgesetzt: alle Daten werden gelöscht.
	Eingabeparameter	
	Vorgang: SYSJOBID	Die Vorgangskennung der Versorgung.
	Ausgabeparameter	
	RetCode	<p>OK</p> <p>ILLEGAL_STATE falls die Transaktion sich nicht im Zustand none befindet.</p> <p>NOT_CONFIGURED falls das Lichtsignalsteuergerät die Versorgung nicht unterstützt. Dies ist insbesondere für die Implementierung von SupplyTransactions relevant, da diese auch blockweise begrenzt sein können.</p> <p>EXISTS_ALREADY falls die angegebene SYSJOBID mit der SYSJOBID der zuletzt initialisierten Transaction übereinstimmt.</p>
101	AddChangeSet	Mit dieser Methode können Änderungen zur Transaktion hinzugefügt werden können.
	Eingabeparameter	
	Vorgang: SYSJOBID	Die Vorgangskennung der Versorgung.
	Objects: BaseObjType[]	<p>Ein Array von Objekten bestehend aus:</p> <ul style="list-style-type: none"> - relativer Path - Daten <p>(<REFPATH_DATA>3</REFPATH_DATA>)</p>
	Ausgabeparameter	

Transaction		
METHOD	Name	Beschreibung
	RetCode	<p>OK</p> <p>ILLEGAL_STATE falls die Änderungen gerade aktiviert werden oder die Transaktion sich im Zustand none oder complete, activationSet oder activating befindet.</p> <p>TOO_MANY falls der Speicher nicht für alle Elemente der Änderungen ausreicht.</p> <p>PARAM_INVALID falls versucht wurde ein Objekt mehrfach hinzuzufügen, oder falls versucht wurde ein Objekt hinzuzufügen das für diese Transaction nicht zulässig ist, z.B. falscher Versorgungsblock.</p> <p>ACCESS_DENIED falls die angegebene SYSJOBID nicht mit der SYSJOBID dieser Transaction übereinstimmt.</p> <p>Hinweis: Bei RetCode != OK werden KEINE Objekte der Transaktion hinzugefügt.</p>
	Flaws: TransactionFlaw[]	Bei RetCode PARAM_INVALID gibt der Array der Schwächen alle Fehler zurück, die beim Versuch die Objekte des ChangeSets der Transaktion hinzuzufügen aufgetreten sind. Die Fehler sind die aufgeführten und entsprechend beschriebenen Messageparts.
102	ReleaseObjects	Mit dieser Methode können Objekte, die bereits aktiviert wurden und nicht mehr genutzt werden, gelöscht werden. Die Methode wird für Teilversorgungen ² benötigt und ist daher optional.
	Eingabeparameter	
	Vorgang: SYSJOBID	Die Vorgangskennung der Versorgung.
	Refs: BaseObjType[]	Ein Array mit Referenzen auf die zu löschenden Objekte.
	Ausgabeparameter	
	RetCode	<p>OK</p> <p>PARAM_INVALID falls eine Objektreferenz unbekannt ist, wird dieser RetCode zurückgegeben.</p> <p>ACCESS_DENIED falls die angegebene SYSJOBID nicht mit der SYSJOBID dieser Transaction übereinstimmt.</p> <p>Hinweis: Bei RetCode != OK werden KEINE Objekte gelöscht.</p>
	Refs: BaseObjType[]	Dieser Rückgabewert beinhaltet eine Liste mit allen unbekannt Objekten.
103	Completed	Diese Methode markiert die Transaction als vollständig. Hinterher kann sich nicht mehr verändert werden.

² Teilversorgungen sind in OCIT-O Lstg V2.0 nicht vorgesehen, die Objektmodellierung berücksichtigt jedoch diese Option auf die Zukunft.

Transaction		
METHOD	Name	Beschreibung
	Eingabeparameter	
	Vorgang: SYSJOBID	Vorgang: SYSJOBID
	Ausgabeparameter	
	RET_CODE	OK ILLEGAL_STATE falls die Transaction sich nicht im Zustand checked befindet. ACCESS_DENIED falls die angegebene SYSJOBID nicht mit der SYSJOBID dieser Transaction übereinstimmt.
104	Activate	Die Anforderung der Aktivierung einer Transaction kann nur in den Zuständen checked oder complete, d.h. nach dem Aufruf der Methode Check bzw. Complete erfolgen. Ein gesetzter Aktivierungszeitpunkt (Zustand im Zustand activationSet) kann durch einen nochmaligen Aufruf der Methode Activate geändert werden. Die Transaktion wird zu einem Zeitpunkt aktiviert. Ist der angegebene Zeitpunkt in der Vergangenheit, werden die zur Transaktion gehörenden Änderungen sofort aktiviert.
	Eingabeparameter	
	Vorgang: SYSJOBID	Vorgang: SYSJOBID
	Zeit: UTC	Der Zeitpunkt der Aktivierung der Änderungen
	Ausgabeparameter	
	RetCode	OK ILLEGAL_STATE falls die Transaction sich nicht im Zustand checked, complete, activationSet befindet. ACCESS_DENIED falls die angegebene SYSJOBID nicht mit der SYSJOBID dieser Transaction übereinstimmt.
105	Abort	OK ILLEGAL_STATE falls die Transaktion sich im Zustand none oder activating befindet.
106	Check	Mit der Methode Check können die Änderungen einer Transaktion daraufhin geprüft werden, ob es möglich ist, sie in das Lichtsignalsteuergeräts zu übernehmen. Hierbei werden die in der Anwenderversorgung standardisierten Daten überprüft.
	Eingabeparameter	
	Vorgang: SYSJOBID	Vorgang: SYSJOBID
	Ausgabeparameter	

Transaction		
METHOD	Name	Beschreibung
	RetCode	<p>OK falls die Änderungen gültig sind und in das Lichtsignalsteuergeräts übernommen werden können.</p> <p>PARAM_INVALID falls der Check fehlgeschlagen ist, mit den Fehlern in der Versorgung als Rückgabewert. Dazu werden die unten definierten Messageparts UndefinedReferenceInObject, DeletedObjectInUse, MissingMandatoryElement, ustAPWertNotWritable und UnspecifiedSupplyError verwendet.</p> <p>ILLEGAL_STATE falls sich die Transaktion nicht im Zustand Receiving befindet.</p> <p>ACCESS_DENIED falls die angegebene SYSJOBID nicht mit der SYSJOBID dieser Transaction übereinstimmt.</p>
	Flaws: TransactionFlaw[]	Bei den RetCodes PARAM_INVALID gibt der Array der Schwächen alle Fehler bzw. notwendigen Änderungen zurück, die bei einer Übernahme der Versorgung notwendig wären. Die Schwächen/Fehler sind die unten aufgeführten und entsprechend beschriebenen Messageparts.
109	SetEventDestination	Setzt das Ziel des OnTransactionStateChanged-Events dieser Transaktion.
	Eingabeparameter	
	Vorgang: SYSJOBID	Die Vorgangskennung der Versorgung.
	EventDst: ZNR_FNR	Zentralen- und Lichtsignalsteuergerätenummer des Gerätes, welches das Event empfangen soll.
	Ausgabeparameter	
	RetCode	<p>OK</p> <p>ACCESS_DENIED falls die angegebene SYSJOBID nicht mit der SYSJOBID dieser Transaction übereinstimmt.</p>

Da beim Aufruf der Methoden AddChangeSet und ReleaseObjects potentiell große Datenmengen transportiert werden, wird für den Aufruf ausschließlich der **TCP Port PNP** verwendet.

Das folgende UML Diagramm illustriert die Korrelation zwischen den Methodenaufrufen und den Zustandsübergängen.

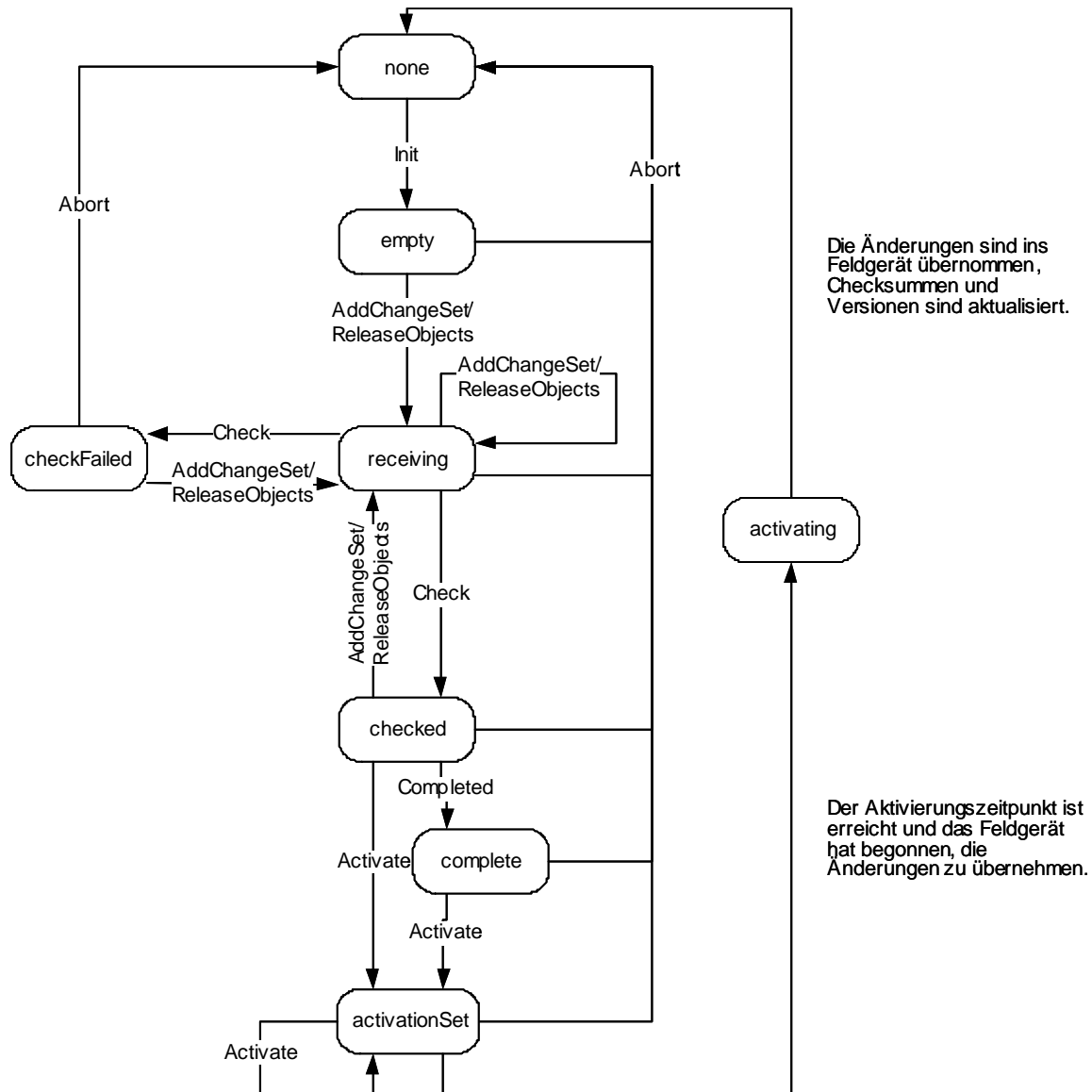


Abbildung 5: Zustandsdiagramm bei Transaktionen

Für jeden Zustand gilt, dass der Aufruf einer unzulässigen Methode mit `ILLEGAL_STATE` quittiert wird und die Transaktion ihren Zustand nicht verändert.

Folgende Ereignisse werden in Meldungen definiert:

- Beim Aufruf von `init` wird die Meldung `VersorgungBeginn` erzeugt mit dem unten definierten Zusatzmeldungsteil `TransactionInitialized`.
- Ein Übergang in den Zustand `none` wird durch eine Meldung `VersorgungEnde` dokumentiert. Der Grund wird ebenfalls durch einen unten spezifizierten Nebemeldungsteil detailliert (`TransactionAborted` oder `TransactionFailed`). Insbesondere `TransactionFailed` kann in jedem Zustand auftreten, z.B. durch einen Netzausfall.

- Der erfolgreiche Aufruf der Methode completed (bzw. activate im Zustand checked) wird durch die Meldung TransactionDefined dokumentiert.

Im Detail:

(MeldungsDegree I: Information, W: Warnung, F: Fehler, S: Schwerer Fehler)

OType	Kurzname	MeldungsDegree	Beschreibung
60300	TransactionInitiali- zed	I	Diese Meldung wird als Zusatzmeldungteil der Meldung VersorgungsBeginn erzeugt, wenn die Methode init aufgerufen wird. Parameter sind der Fully Qualified Domain Name (FQDN) bzw. die IP-Adresse (wenn FQDN nicht aufgelöst werden kann) des Lichtsignalsteuergeräts des Aufrufers (RemoteDevice)
60301	TransactionDefined	I	Diese Meldung wird erzeugt, wenn die Transaction sich im Zustand checked befindet und die Methode Completed aufgerufen wird.
60318	TransactionActiva- tionRequest	I	Diese Meldung wird erzeugt, wenn die Aktivierung einer Transaktion angefordert wurde, d. h. die Methode Activate aufgerufen wurde. Parameter ist der Zeitpunkt der angeforderten Aktivierung.
60302	TransactionAborted	I	Diese Meldung wird als Zusatzmeldungteil der Meldung VersorgungsEnde erzeugt, falls die Methode abort aufgerufen wird.
60303	TransactionFailed	F	Diese Meldung wird als Zusatzmeldungsteil der Meldung VersorgungsEnde erzeugt, falls aus irgendeinem undefinierten Grund die Transaktion fehlgeschlagen ist.

Die Zuordnung der Meldungen erfolgt über die SYSJOBID.

Folgende Meldungsteile werden als Zusatzmeldungsteil(e) verwendet und dienen zusätzlich als Rückgabewerte der Funktion Transaction.Check():

OType	Kurzname	MeldungsDegree	Beschreibung
60309	UnspecifiedFlaw	W	Falls eine Versorgung eine Schwäche enthält, so wird diese Meldung als Zusatzmeldungsteil der Meldung VersorgungsEnde erzeugt. Parameter sind das Objekt, das die Schwäche enthält und ein Text mit einer Beschreibung.
60310	UnspecifiedSupplyError	F	Falls eine Versorgung einen Fehler enthält, so wird diese Meldung als Zusatzmeldungsteil der Meldung VersorgungsEnde erzeugt. Parameter sind das Objekt, das die Schwäche enthält und ein Text mit einer Beschreibung.

Folgende Meldungsteile werden nicht in Archive eingetragen, sondern dienen nur als Rückgabewerte der Funktion Transaction.Check():

OType	Kurzname	MeldungsDegree	Beschreibung
60304	UndefinedReferenceInObject	F	Dieser Meldungsteil dient dazu, Fehler in der Versorgung zu melden. Es gibt 2 Parameter: eine Referenz auf das Objekt in dem der Fehler aufgetreten ist und die undefinierte Objektreferenz.
60305	DeletedObjectInUse	F	Dieser Meldungsteil dient dazu, Fehler in der Versorgung zu melden. Er zeigt an, dass versucht wurde, ein Objekt zu löschen, das noch in Benutzung ist. Parameter ist eine Referenz auf das Objekt, das gelöscht werden sollte.
60306	MissingMandatoryElement	F	Dieser Meldungsteil dient dazu, Fehler in der Versorgung zu melden. Er zeigt an, dass ein notwendiges Element nicht versorgt wurde. Dies können im Falle einer Supply-Transaction (siehe unten) die OCIT-I Versionen der geänderten Blocks oder Standardelemente (z.B. Standardtages- oder Wochenplan) sein. Parameter ist eine Referenz auf das fehlende Element.
60308	ObjectNotInBlock	F	Dieser Meldungsteil dient dazu, Fehler in der Versorgung

OType	Kurzname	MeldungsDegree	Beschreibung
			zu melden und wird erzeugt, falls versucht wurde ein Objekt zu erzeugen, das nicht im zu versorgenden Block liegt. Parameter ist eine Referenz auf dieses Objekt.
60317	APWertWriteNotPossible	F	Dieser Meldungsteil dient dazu, Fehler in der Versorgung zu melden. Er wird erzeugt, falls versucht wird, einen AP-Wert zu ändern, der nicht schreibbar ist. Parameter ist eine Referenz auf den APWert.
60320	DuplicateObject	F	Dieser Meldungsteil dient dazu, Fehler in der Versorgung zu melden. Er wird erzeugt, falls versucht wird, ein Objekt mehrmals zu einer Transaktion hinzuzufügen. Parameter ist eine Referenz auf dieses Objekt.

3.1.2 Objekt SupplyTransaction (Versorgungstransaktion)

Zur Übertragung und Aktivierung von Anwenderversorgungsdaten wird im Lichtsignalsteuergerät ein Objekt implementiert, die SupplyTransaction, die diese Funktionalität zur Verfügung stellt. Grundsätzlich muss dazu ein Versorgungsdatenpuffer vorgesehen werden, in dem die mit der SupplyTransaction übertragenen Daten zwischengespeichert werden. Der Puffer muss ausreichend dimensioniert sein, um eine Neuversorgung aller Blöcke des Lichtsignalsteuergeräts zwischenspeichern zu können.

Die Basedomain der SupplyTransaction ist die Transaction. Die Abbildung 4 (Zustandsdiagramm bei Transaktionen) gilt auch für die Versorgungstransaktion.

Die SupplyTransaction erweitert die Transaction um Funktionalität zur Versorgung der Blöcke der Anwenderversorgung.

Hinweis: In OCIT-O Lstg V2.0 erfolgt die Anwenderversorgung immer blockweise, d. h. sie ist mit allen Objekten eines der 4 Versorgungsdatenblöcke durchzuführen (siehe Pkt. 2.3.1 und 2.3.3).

Für die Aktivierung einer Versorgungstransaktion gelten besondere Bedingungen:

- Die OCIT-I Versionsinformationen (Version und Checksumme, siehe Pkt. 3.2.2) eines Blocks müssen immer geändert werden, wenn sich ein Datum innerhalb des Blocks ändert.
- Im Block „Daten mit Netzbezug“ müssen Standardelemente (Standardtagesplan und Standardwochenplan) immer vorhanden sein, sonst wird die Versorgung abgelehnt.
- Falls alle Daten eines Blocks der Anwenderversorgung versorgt werden (Standardmethode) werden alle zum Block gehörenden Objekte vor der Aktivierung gelöscht. Die SupplyTransaction erkennt diese Objekte an der Blockkennung (VDArt) der zu versorgenden Objekte.

SupplyTransaction (1:711)

SupplyTransaction		
METHOD	Name	Beschreibung
120	InitSupplyTransaction	<p>Die Transaktion wird zurückgesetzt: alle Daten werden gelöscht.</p> <p>Die Methode InitSupplyTransaction enthält im Eingabeparameter die Information darüber, von welcher Art die Versorgung ist. Für jeden zu versorgenden Block wird dessen VDArt im Eingabeparameter übergeben. Soll auf diese Art blockweise versorgt werden, so wird er komplett versorgt. Dazu werden vor der Aktivierung alle Objekte des Blocks gelöscht und mit dem Inhalt der Transaktion überschrieben.</p> <p>Sind der Array von VDarten leer, so soll eine Teilversorgung³ stattfinden. Steuergeräte, die diese Versorgungsart nicht unterstützen, beantworten den Methodenaufruf mit einer Fehlermeldung. Bei einer Teilversorgung müssen überschüssige Objekte explizit durch Aufruf der Methode ReleaseObjects gelöscht werden.</p> <p><u>Hinweis:</u> In diesem Fall entspricht der Aufruf dem Aufruf von Init (Methode 100) der Basedomain Transaction 1:710.</p>
	Eingabeparameter	
	Vorgang: SYSJOBID	Die Vorgangskennung der Versorgung.
	Blocks.Anzahl	Anzahl folgender Blocks

³ Teilversorgungen sind in OCIT-O Lstg V2.0 nicht vorgesehen, die Objektmodellierung berücksichtigt jedoch diese Option auf die Zukunft.

SupplyTransaction		
METHOD	Name	Beschreibung
	Blocks: VDArt[0..3]	Ein Array mit allen zu versorgenden Blocks. Falls Blocks die Länge 0 hat, wird eine Teilversorgung durchgeführt. Jede VDArt darf nur einmal enthalten sein!
	Ausgabeparameter	
	RetCode	OK ILLEGAL_STATE falls die Transaktion sich nicht im Zustand none befindet. NOT_CONFIGURED falls Teilversorgung angefordert wurde und das Lichtsignalsteuergerät diese nicht unterstützt. EXISTS_ALREADY falls die angegebene SYSJOBID mit der SYSJOBID der zuletzt initialisierten Transaction übereinstimmt.
121	ReadVD	Mit der Methode ReadVD können die Versorgungsdaten eines Blockes auf einmal gelesen werden. Die Methode benötigt keine Transaktionsinstanz und liefert die aktuell aktiven Versorgungsobjekte.
	Eingabeparameter	
	VDArtFilter.Anzahl	Anzahl folgender VDArt Elemente
	VDArt:VDArtFilter[0...3]	Kennung der Blöcke deren Versorgungsdaten geliefert werden sollen. Falls VDArtFilter die Länge 0 hat, sind alle Blöcke gemeint. Jede VDArt darf nur einmal enthalten sein!
	Ausgabeparameter	
	RetCode	OK PARAM_INVALID falls unbekannter VDArtFilter
	VD : VersorgbaresObjekt[]	Liste versorgbarer Objekte . Diese werden Lichtsignalsteuergeräterelativ mit Pfad und Daten übertragen (s.a. Transaction:: AddChangeSet). Der Inhalt der Versorgungsobjekte ist der mit welchem das Lichtsignalsteuergerät zum Zeitpunkt des Aufrufs arbeitet.
122	ReadVDExt	Mit der Methode ReadVDExt können die Versorgungsdaten eines Blockes, die LSAVersionen sowie die Gesamtversion auf einmal gelesen werden. Die Methode benötigt keine Transaktionsinstanz und liefert die aktuell aktiven Versorgungsobjekte.
	Eingabeparameter	
	VDArtFilter.Anzahl	Anzahl folgender VDArt Elemente

SupplyTransaction		
METHOD	Name	Beschreibung
	VDart:VDartFilter[0...3]	Kennung der Blöcke deren Versorgungsdaten geliefert werden sollen. Falls VDartFilter die Länge 0 hat, sind alle Blöcke gemeint. Jede VDart darf nur einmal enthalten sein!
	Ausgabeparameter	
	RetCode	OK PARAM_INVALID falls unbekannter VDartFilter
	VD : VersorgbaresObjekt[]	Liste versorgbarer Objekte . Diese werden Lichtsignalsteuergeräterelativ mit Pfad und Daten übertragen (s.a. Transaction:: AddChangeSet). Der Inhalt der Versorgungsobjekte ist der mit welchem das Lichtsignalsteuergerät zum Zeitpunkt des Aufrufs arbeitet.
	GesamtVersion	Gesamtversion
	LsaVersion[]	Liste der LSA-Versionsobjekte
123	ReadVDExtPlus	Mit der Methode ReadVDExt können die Versorgungsdaten eines Blockes, die LSAVersionen sowie die Gesamtversion auf einmal gelesen werden. Die Methode benötigt keine Transaktionsinstanz und liefert die aktuell aktiven Versorgungsobjekte.
	Eingabeparameter	
	VDartFilter.Anzahl	Anzahl folgender VDart Elemente
	VDart:VDartFilter[0...3]	Kennung der Blöcke deren Versorgungsdaten geliefert werden sollen. Falls VDartFilter die Länge 0 hat, sind alle Blöcke gemeint. Jede VDart darf nur einmal enthalten sein!
	Ausgabeparameter	
	RetCode	OK PARAM_INVALID falls unbekannter VDartFilter
	VD : VersorgbaresObjekt[]	Liste versorgbarer Objekte . Diese werden Lichtsignalsteuergeräterelativ mit Pfad und Daten übertragen (s.a. Transaction:: AddChangeSet). Der Inhalt der Versorgungsobjekte ist der mit welchem das Lichtsignalsteuergerät zum Zeitpunkt des Aufrufs arbeitet.
	GesamtVersion	Gesamtversion
	LsaVersionPlus[]	Liste der LSA-Versionsobjekte mit SYSJOBID.
0	Standardmethode Get	
	Ausgabeparameter	

SupplyTransaction		
METHOD	Name	Beschreibung
	RetCode	OK
	Vorgang: SYSJOBID	Die Versorgungskennung der laufenden Transaction. NULLVALUE falls sich die Versorgung im Zustand none befindet.
	EventDst: ZNR_FNR	Zentralen- und Lichtsignalsteuergerätenummer des Gerätes an welches diese Transaction das OnTransactionState-Changed-Event sendet.
	CompletionTime: UTC	Der Zeitpunkt, zu dem die Transaction als complete markiert wurde. Sonst NULLVALUE.
	ActivationTime: UTC	Der aktuelle Aktivierungszeitpunkt, falls gesetzt. Sonst NULLVALUE.
	State	Der aktuelle Zustand der Transaction.
	Blocks: VDArt[]	Der Parameter Blocks, der bei Get zurückgegeben wird, ist derselbe wie derjenige, der mit InitSupplyTransaction gesetzt wird.

Die Versorgungsänderungen werden mit einem speziellen Zusatzmeldungsteil, der den genauer spezifiziert eingetragen:

(MeldungsDegree I: Information, W: Warnung, F: Fehler, S: Schwerer Fehler)

OType	Kurzname	MeldungsDegree	Beschreibung
60311	SupplyDefined	I	Diese Meldung wird als Zusatzmeldungsteil der Meldung TransactionDefined erzeugt. Parameter ist ein VDArt-Array der Blöcke für die eine neue Versorgung definiert wurde.
60312	SupplyActivated	I	Diese Meldung wird als Zusatzmeldungsteil der Meldung VersorgungsEnde erzeugt, wenn die Änderungen einer Versorgungstransaktion in die Geräteversorgung übernommen wurden.
60314	CurrentBlockVersion	I	Diese Meldung wird als Zusatzmeldungsteil der Meldung SupplyVersionChanged für jeden geänderten Block erzeugt. Parameter sind die VDVersion und die LsaVersion des Blocks.

OType	Kurzname	MeldungsDegree	Beschreibung
			Anmerkung: Die SYSJOBID des Vorgangs wird in der SYSJOBID der Meldung bereits dokumentiert. Daher wird in in diesem Fall nicht die LsaVersionPlus genutzt.
60313	CurrentFeldgeraete-Version	I	Diese Meldung wird als Zusatzmeldungsteil der Meldung SupplyVersionChanged erzeugt. Parameter ist die GesamtVersion der Versorgung.
60319	SupplyVersionChanged	I	<p>Diese Meldung wird erzeugt wenn sich VDVersion und/oder LsaVersion mindestens eines Versorgungsblocks geändert haben (bei Anwender- oder Hersteller-versorgung).</p> <p>Die Meldung wird nur in das Versorgungsarchiv, nicht in das Standardmeldearchiv, eingetragen.</p> <p>Als Zusatzmeldungsteile von SupplyVersionChanged werden CurrentBlockVersion (für jeden geänderten Block) und CurrentFeldgeraeteVersion eingetragen.</p>

3.1.3 Objekt TransferParameterBlock

Das Objekt TransferParameterblock dient zum blockweisen Schreiben von AP-Werten. Es ist als optional deklariert.

Um mehrere AP-Werte oder komplex strukturierte AP-Werte im Lichtsignalsteuergerät zu setzen (so weit sie setzbar sind), wird das Objekt TransferParameterBlock definiert. Dieses Objekt stellt analog wie die Versorgungstransaktion für versorgbare Objekte diese Funktionalität für AP-Werte bereit.

Grundsätzlich muss dazu ein AP-Blockpuffer vorgesehen werden, in dem die mit TransferParameterBlock übertragenen Daten zwischengespeichert werden. Der Puffer muss entsprechend den vorgesehenen Anwendungen dimensioniert sein.

Die Basedomain des TransferParameterBlock ist die Transaction.

Zur effizienten Übertragung der Änderungen bietet TransferParameterBlock eine zusätzliche Methode AddPrefixedChangeSet an, die einen zusätzlichen Pfad Parameter besitzt, der allen nachfolgenden AP-Werten vorangestellt wird (getrennt durch einen Punkt).

Für jedes Schreiben der AP-Werte wird eine entsprechende Kurzmeldung im Standardmeldearchiv eingetragen und ausführliche Informationen in das Versorgungsarchiv.

TransferParameterBlock ist abgeleitet von Transaction. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

TransferParameterBlock (1:712)

TransferParameterBlock		
METHOD	Name	Beschreibung
150	AddPrefixedChangeSet	Mit dieser Methode können Änderungen zur Transaktion hinzugefügt werden können.
	Eingabeparameter	
	Vorgang: SYSJOBID	Vorgang: SYSJOBID
	Prefix: ANYPATH	Ein ANYPATH, der den Namen der nachfolgend referenzierten APWerte vorangestellt wird.
	Objects: BaseObjType[]	Ein Array von Objekten bestehend aus: <ul style="list-style-type: none"> - relativer Path - Daten (<REFPATH_DATA>3</REFPATH_DATA>)
	Ausgabeparameter	
	RetCode	OK ILLEGAL_STATE falls die Änderungen gerade aktiviert werden oder die Transaktion sich im Zustand none befindet. TOO_MANY falls der Speicher nicht für alle Elemente der Änderungen ausreicht. ACCESS_DENIED falls die angegebene SYSJOBID nicht mit der SYSJOBID dieser Transaction übereinstimmt. PARAM_INVALID falls versucht wurde ein Objekt hinzuzufügen das für diese Transaction nicht zulässig ist, z.B. Objekt ist kein APWert. Hinweis: Bei RetCode != OK werden KEINE Objekte der Transaktion hinzugefügt.
	Flaws: TransactionFlaw[]	Bei RetCode PARAM_INVALID gibt der Array der Schwächen alle Fehler zurück, die beim Versuch die Objekte des ChangeSets der Transaktion hinzuzufügen aufgetreten sind. Die Fehler sind die aufgeführten und entsprechend beschriebenen Messageparts.
106	Check	Die Methode der BASEDOMAIN wird hier übernommen. Zusätzlich jedoch schlägt Check fehl, wenn versucht wurde, einen AP-Wert zu ändern, der nicht schreibbar ist.
	Eingabeparameter	
	Vorgang: SYSJOBID	Vorgang: SYSJOBID

TransferParameterBlock		
METHOD	Name	Beschreibung
	Ausgabeparameter	
	RetCode	Siehe RetCode der BASEDOMAIN Transaction.
0	Standardmethode Get	
	Ausgabeparameter	
	RetCode	OK
	Vorgang: SYSJOBID	Die Versorgungskennung der laufenden Transaction. NULLVALUE falls sich die Versorgung im Zustand none befindet.
	EventDst: ZNR_FNR	Zentralen- und Lichtsignalsteuergerätenummer des Gerätes an welches diese Transaction das OnTransactionState-Changed-Event sendet.
	CompletionTime: UTC	Der Zeitpunkt, zu dem die Transaction als complete markiert wurde. Sonst NULLVALUE.
	ActivationTime: UTC	Der aktuelle Aktivierungszeitpunkt, falls gesetzt. Sonst NULLVALUE.
	State	Der aktuelle Zustand der Transaction.

Zur Dokumentation der Änderung der AP-Werte werden bereits bei der Transaction definierten Meldungen in das Standardmeldearchiv eingetragen.

Zusätzlich stehen folgende Meldungen zur Verfügung, die nur in das Versorgungsarchiv, nicht in das Standardmeldearchiv eingetragen werden.

OType	Kurzname	Meldungs-Degree	Beschreibung
60315	APWertChange-Requested	I	Diese Meldung wird erzeugt, wenn das Activate des TransferParameterBlocks aufgerufen wird. Parameter sind <ul style="list-style-type: none"> • Referenz auf den APWert • Alter Wert • Neuer Wert
60316	APWertChangeCommitted	I	Diese Meldung wird erzeugt, wenn die Änderungen aus der Meldung APWertChangeRequested übernommen wurden. Parameter sind Programmnummer, Programmzeit (TX) und Phasennummer (PH) vom Zeitpunkt der APWert-Übernahme.

60317	APWertWriteNot-Possible	F	Dieser Meldungsteil dient dazu, Fehler in der Versorgung zu dokumentieren. Er wird erzeugt, falls versucht wird, einen AP-Wert zu ändern, der nicht schreibbar ist. Parameter ist eine Referenz auf den APWert.
-------	-------------------------	---	---

3.2 Versionierung der Versorgungsdaten

Mittels des in OCIT-I VD und OCIT-O vorgesehenen Versionierungsverfahrens können Versorgungsänderungen durch Kontrolle von Versionsnummern und Checksummen nachvollzogen werden (auch als Rückdokumentation bezeichnet). Einem Versorgungswerkzeug werden mit dem Versionierungsverfahren alle Möglichkeiten gegeben, sowohl Veränderungen als auch den Erhalt der eigentlichen verkehrstechnischen Daten zu analysieren. Zusätzlich gibt es eine Möglichkeit zum Lesen von Versorgungsdaten des Lichtsignalsteuergeräts.

Die Dokumentation der Versorgung (was gehört wie zusammen, z. B. Anwenderversorgung und Herstellerversorgung) ist nicht Bestandteil der Spezifikation von OCIT-O Lstg V2.0, sondern eine Eigenschaft des Versorgungswerkzeugs. Die Registratur der verschiedenen Versionsstände, ihrer Kombinationen, lokalen Änderungen und der technischen Freigaben muss der Betreiber organisieren.

Entsprechend der blockweisen Versorgung wird auch blockweise versioniert. Versionierungsdaten werden sowohl im OCIT-I VD Server als auch im Lichtsignalsteuergerät gespeichert. Im Lichtsignalsteuergerät gespeicherte Daten können vom Server gelesen werden.

Versionierungsdaten, die vom OCIT-I VD Server erzeugt und im Lichtsignalsteuergerät gespeichert werden sollen, werden zusammen mit den Versorgungsdaten mittels der Supply-Transaction übertragen. Versionierungsdaten, die im Lichtsignalsteuergerät erzeugt werden, werden mit jeder Versorgungsänderung (lokal oder zentral) aktualisiert.

Das Lichtsignalsteuergerät bietet mit den gespeicherten, erzeugten und auslesbaren Versionierungsdaten die Grundlage für die Verwaltung und Kontrolle der Versionen und Daten von Geräteversorgungen.

Schema der Versorgungsdatenblöcke und Versionierungsdaten:

In OCIT-I VD standardisierte Versorgungsdaten, herstellerübergreifend versorgbar und auslesbar Anwenderversorgung				In OCIT-I VD teilweise stand. Versorgungsdaten, nur proprietär versorgbar und auslesbar Herstellerversorgung			
Verkehrstechnik				Gerätetechnik		Sicherheitstechnik	
Verkehrstechnische Grunddaten / Festzeit	Daten mit Netzbezug	VA-Steuerverfahren	VA-Parameter	OCIT-I VD Versorgungsdaten	Proprietäre Daten	OCIT-I VD Sicherheitsdaten	Proprietäre Daten
.....
.....
.....
OCIT-O Checksum. Server 4)	OCIT-O Checksum. Server 4)	OCIT-O Checksum. Server 4)	OCIT-O Checksum. Server 4)	Blau umrandet: Versionierungsdaten des OCIT-I VD Servers, die in OCIT-I Komponenten verwaltet werden.			
OCIT-I Version 1)	OCIT-I Version 1)	OCIT-I Version 1)	OCIT-I Version 1)	Hersteller-Version 2)		Hersteller-Version 2)	
OCIT-I Checksumme 1)	OCIT-I Checksumme 1)	OCIT-I Checksumme 1)	OCIT-I Checksumme 1)	Hersteller-Checksumme 2)		Hersteller-Checksumme 2)	
Auftraggeber 1)	Auftraggeber 1)	Auftraggeber 1)	Auftraggeber 1)				
OCIT-O Checksum. Gerät 3)	OCIT-O Checksum. Gerät 3)	OCIT-O Checksum. Gerät 3)	OCIT-O Checksum. Gerät 3)	Hersteller-Checksumme Gerät 3)		Hersteller-Checksumme Gerät 3)	
Build-Nr. 3)	Build-Nr. 3)	Build-Nr. 3)	Build-Nr. 3)	Build-Nr. 3)		Build-Nr. 3)	
Session-ID 5)	Session-ID 5)	Session-ID 5)	Session-ID 5)				
Zeitstempel Übertragung beendet 3)	Zeitstempel Übertragung beendet 3)	Zeitstempel Übertragung beendet 3)	Zeitstempel Übertragung beendet 3)	Schwarz umrandet: Versionierungsdaten, die im Lichtsignalsteuergerät verwaltet werden (OCIT-O Objekt Versionierung)			
Zeitstempel Aktivierung 3)	Zeitstempel Aktivierung 3)	Zeitstempel Aktivierung 3)	Zeitstempel Aktivierung 3)				
Fully Qualified Domain Name 3)	Fully Qualified Domain Name 3)	Fully Qualified Domain Name 3)	Fully Qualified Domain Name 3)				
OCIT-O Checksum. Gerät Gesamt 3)							
Build-Nr. Gesamt 3)							

- 1) Vom OCIT-I VIAP gebildet und im Lichtsignalsteuergerät gespeichert.
- 2) Vom Herstellerwerkzeug gebildet und im Lichtsignalsteuergerät gespeichert.
- 3) Im Lichtsignalsteuergerät erzeugt und gespeichert.
- 4) Im OCIT-I VD Server erzeugt und gespeichert.
- 5) Im OCIT-I VD Server erzeugt und im Lichtsignalsteuergerät gespeichert.

Tabelle 1: Beschreibung der Versionierungsdaten

Versionierungsdaten	Quelle	Zeitpunkt der Entstehung	Speicherorte	Beschreibung
Versionierungsdaten des OCIT-I VD Servers, die in OCIT-Komponenten verwaltet werden:				
OCIT-O Checksumme Server Block 1 bis Block 4	OCIT-I VD Server	Während einer Versorgungs- transaktion.	OCIT-I VD Server OCIT-I VIAP	<i>OCIT-O Checksummen Server</i> werden im Server über die ins Lichtsignalsteuergerät zu übertragenen Versorgungsdaten mittels eines standardisierten Verfahrens ⁴ gebildet. <u>Option:</u> Bildung der <i>OCIT-O Checksummen Server</i> im VD-Server nach Abschluss der Planung auf Anforderung des Planungstools. Damit kann bereits zum Planungszeitpunkt, ohne die Versorgung in das Lichtsignalsteuergerät zu übertragen, jene Checksumme erzeugt werden, die nach einer Versorgung auch im Lichtsignalsteuergerät erzeugt werden soll.
Versionierungsdaten, die im Lichtsignalsteuergerät verwaltet werden (OCIT-O Objekt Versionierung)				
OCIT-I Version Block 1 bis Block 4	Planer	Planungs- vorgang	OCIT-I VIAP OCIT-O Objekt Versionierung	Für jeden Block der Anwenderversorgung werden vom Planer am OCIT-I VIAP die <i>OCIT-I Versionsbezeichnungen</i> vergeben, beim Versorgungsvorgang ins Lichtsignalsteuergerät übertragen und dort gespeichert. Diese Informationen werden im Lichtsignalsteuergerät nicht verändert und können über OCIT-O ausgelesen werden.
OCIT-I Checksumme Block 1 bis Block 4	OCIT-I VIAP	Planungs- vorgang	OCIT-I VIAP OCIT-O Objekt Versionierung	Für jeden Block der Anwenderversorgung werden vom OCIT-I VIAP die <i>OCIT-I Checksummen</i> berechnet, beim Versorgungsvorgang ins Lichtsignalsteuergerät übertragen und dort gespeichert. Diese Informationen werden im Lichtsignalsteuergerät nicht verändert und können über OCIT-O ausgelesen werden.

⁴ Ziel ist es, die OCIT-O Checksumme Server und OCIT-O Checksumme Gerät so zu bilden, dass bei gleichen Versorgungsdaten die gleiche Checksumme entsteht.

Versionierungsdaten	Quelle	Zeitpunkt der Entstehung	Speicherorte	Beschreibung
Auftraggeber Block 1 bis Block 4	Planer	Planungsvorgang	OCIT-I VIAP OCIT-O Objekt Versionierung	Für jeden Block der Anwenderversorgung werden vom Planer am OCIT-I VIAP die Namen der <i>Auftraggeber</i> (String 256 Zeichen) eingetragen, ins Lichtsignalsteuergerät übertragen und dort gespeichert. Diese Informationen werden im Lichtsignalsteuergerät nicht verändert und können über OCIT-O ausgelesen werden.
OCIT-O Checksumme Gerät Block 1 bis Block 4	Lichtsignalsteuergerät	Nach Aktivierung der Versorgung eines Blocks oder bei lokalen Versorgungsänderungen.	OCIT-O Objekt Versionierung	<i>OCIT-O Checksummen Gerät</i> werden über die im Lichtsignalsteuergerät abgespeicherten Versorgungsdaten mittels des standardisierten Verfahrens gebildet. Die Checksummen können über OCIT-O ausgelesen werden.
Build-Nummer Block 1 bis Block 4, Block Gerätetechnik, Block Sicherheitstechnik	Lichtsignalsteuergerät	Nach Aktivierung der Versorgung eines Blocks oder bei lokalen Versorgungsänderungen.	OCIT-O Objekt Versionierung	Im Steuergerät wird bei jeder Aktivierung eines Versorgungsblocks eine neue, den Blöcken zugeordnete <i>Build-Nummer</i> erzeugt, indem ein Zähler hochgezählt wird. Eine Änderung der Build-Nummer signalisiert dem Versorger eine Änderung der Versorgungsdaten im Gerät. Ein Rücksetzen ist nicht vorgesehen, der Zähler läuft einfach über. Build-Nummern können von den Versorgungswerkzeugen über OCIT-O ausgelesen werden.
Session-ID Block 1 bis Block	Lichtsignalsteuergerät	Während eines Versorgungsvorgangs	OCIT-O Objekt Versionierung	Die während einer Versorgung eines Blocks verwendete <i>Session-ID</i> (entspricht der im OCIT-I VD Server erzeugten OCIT-O Job-ID) wird im Lichtsignalsteuergerät gespeichert und kann von den Versorgungswerkzeugen über OCIT-O ausgelesen werden.
Zeitstempel Übertragung beendet Block 1 bis Block	Lichtsignalsteuergerät	Nach Abschluss der Übertragung	OCIT-O Objekt Versionierung	Der <i>Zeitstempel Übertragung beendet</i> enthält die Uhrzeit beim Ende der Übertragung der Versorgungsdaten eines Blocks. Er wird im Lichtsignalsteuergerät gespeichert und kann über OCIT-O ausgelesen werden.
Zeitstempel	Lichtsignal-	Nach Aktivierung der	OCIT-O Objekt	Der <i>Zeitstempel Aktivierung</i> enthält die Uhrzeit zu der die aktuell gültige Versorgung eines

Versionierungsdaten	Quelle	Zeitpunkt der Entstehung	Speicherorte	Beschreibung
Aktivierung Block 1 bis Block 4	steuergerät	Versorgung eines Blocks	Versionierung	Blocks von einem OCIT-I VD Server aus aktiviert wurde. Er wird im Lichtsignalsteuergerät gespeichert und kann über OCIT-O ausgelesen werden.
Fully Qualified Domain Name Block 1 bis Block 4	Versorgungswerkzeug	Nach Aktivierung der Versorgung eines Blocks	OCIT-O Objekt Versionierung	Fully Qualified Domain Name (FQDN) oder falls der Reverse Lookup fehlschlägt die IP-Adresse (in dotted decimal notation) von der aus die letzte erfolgreiche Versorgung dieser Art übernommen wurde. Im Fall der direkten lokalen Versorgung mittels nicht IP- basiertem Herstellertool ist der Name des Geräts selbst einzutragen. Der FQDN wird im Lichtsignalsteuergerät gespeichert und kann über OCIT-O ausgelesen werden.
Hersteller-Version Block Gerätetechnik Block Sicherheitstechnik	Spezialist des Geräteherstellers	Planungsvorgang	Herstellerwerkzeug OCIT-O Objekt Versionierung	Für jeden dieser Blöcke werden vom Spezialisten am Herstellerwerkzeug die Versionsbezeichnungen vergeben, ins Lichtsignalsteuergerät übertragen und dort gespeichert. Diese Informationen werden im Lichtsignalsteuergerät nicht verändert und können über OCIT-O ausgelesen werden.
Hersteller-Checksumme Block Gerätetechnik, Block Sicherheitstechnik	Herstellerwerkzeug	Nach Aktivierung der Versorgung eines Blocks oder bei lokalen Versorgungsänderungen.	Herstellerwerkzeug OCIT-O Objekt Versionierung	Für jeden dieser Blöcke werden vom Herstellerwerkzeug die <i>Hersteller-Checksummen</i> mittels des standardisierten oder eines proprietären Verfahrens gebildet, ins Lichtsignalsteuergerät übertragen und dort gespeichert. Diese Informationen werden im Lichtsignalsteuergerät nicht verändert und können über OCIT-O ausgelesen werden.
Hersteller-Checksumme Gerät Block Gerätetechnik, Block Sicherheitstechnik	Lichtsignalsteuergerät	Nach Aktivierung der Versorgung eines Blocks oder bei lokalen Versorgungsänderungen.	OCIT-O Objekt Versionierung	<i>Hersteller-Checksummen Gerät</i> werden über die im Lichtsignalsteuergerät abgespeicherten Versorgungsdaten mittels des standardisierten oder eines proprietären Verfahrens gebildet. Die Checksummen können über OCIT-O ausgelesen werden.

Versionierungsdaten	Quelle	Zeitpunkt der Entstehung	Speicherorte	Beschreibung
OCIT-O Checksumme Gerät gesamt	Lichtsignalsteuergerät	Nach Aktivierung der Versorgung eines Blocks oder bei lokalen Versorgungsänderungen.	OCIT-O Objekt Versionierung	Die <i>OCIT-O Checksumme gesamt</i> wird über alle im Lichtsignalsteuergerät abgespeicherten Versorgungsdaten mittels des standardisierten oder eines proprietären Verfahrens gebildet. Die Checksumme kann über OCIT-O ausgelesen werden.
Build-Nummer gesamt	Lichtsignalsteuergerät	Nach Aktivierung der Versorgung eines Blocks oder bei lokalen Versorgungsänderungen.	OCIT-O Objekt Versionierung	Im Steuergerät wird bei jeder Aktivierung eines der Versorgungsblöcke eine neue <i>Build-Nummer gesamt</i> erzeugt, in dem ein Zähler hochgezählt wird. Eine Änderung der Build-Nummer signalisiert dem Versorger eine Änderung der Versorgungsdaten im Gerät. Ein Rücksetzen ist nicht vorgesehen, der Zähler läuft einfach über. Die Build-Nummer gesamt kann über OCIT-O ausgelesen werden.

Hinweis: Es wird empfohlen, bei der Darstellung von Checksummen an Oberflächen die im OCIT-O Protokoll für Meldungen festgelegte Darstellungsform zu verwenden.

Die Versionierungsdaten geben Auskunft darüber:

- welche Version versorgt ist (Versionen),
- ob die zur Version zugehörigen und im Versorgungswerkzeug gespeicherten Daten jene sind, die ins Lichtsignalsteuergerät übertragen wurden (OCIT-I, OCIT-O und Hersteller-Checksummen),
- das geändert wurde (Build-Nr.),
- welcher Block lokal oder auf anderen Wegen verändert wurde (OCIT-O Checksummen),
- wann übertragen wurde (Zeitstempel Übertragung beendet)
- wann geändert wurde (Zeitstempel Aktivierung) und
- von wo aus geändert wurde (Fully Qualified Domain Name, Session-ID).

Hinweis: Es wird empfohlen bei jeder Aktivierung einer Versorgung alle Versionierungsdaten aus dem Lichtsignalsteuergerät und dem OCIT-I VD Server zu lesen und in der Versionsverwaltung des VIAPs abzulegen. Was geändert wurde, kann durch blockweises Auslesen der im Lichtsignalsteuergerät tatsächlich vorhandenen Anwenderversorgung gefunden werden.

3.2.1 Standard-Verfahren zur Checksummenbildung

Die „OCIT-O Checksummen Gerät“ sollen in allen 4 Versorgungsblöcken so gebildet werden, dass bei gleichen Versorgungsdaten in allen OCIT konformen Lichtsignalsteuergeräten die gleichen Checksummen entstehen. Um dies zu erreichen, wird für die Checksummenbildung ein standardisiertes Verfahren verwendet.

Es wird empfohlen auch im OCIT-I VD-Server dieses Verfahren anzuwenden, um nach Abschluss der Planung die „OCIT-O Checksumme Server“ zu bilden. Damit kann bereits zum Planungszeitpunkt, ohne die Versorgung in das Lichtsignalsteuergerät zu übertragen, jene Checksumme erzeugt werden, die nach einer Versorgung auch im Lichtsignalsteuergerät erzeugt werden soll. Man erreicht dadurch eine sehr hohe Sicherheit, dass die vom Planer gewollte Versorgung auch tatsächlich im Lichtsignalsteuergerät umgesetzt wird.

Die Vorschrift zur Checksummenberechnung besteht aus drei Komponenten:

- Festlegung, in welcher Reihenfolge die Versorgungsdaten der Versorgungsobjekte zur Bildung der Checksummen gelesen werden (alle Elemente der Versorgungsobjekte in aufsteigender Reihenfolge, siehe dazu die Festlegungen in 3.3 zu „sortiert und gelesen“)
- dass die Versorgungsdaten vom VD-Server in dieser Reihenfolge übertragen werden, und
- der Vorschrift zur Berechnung der Checksumme (mit Standardalgorithmus SHA-1, siehe Dokument OCIT-O Basis).

Hinweis: Die „OCIT-O Checksummen Gerät“ wird nicht aus den Daten im Versorgungsdatenpuffer gebildet, sondern aus den „echten“ Versorgungsdaten, die entweder nach der Aktivierung einer Versorgung oder bei lokalen Änderungen, abgespeichert werden. Der Hersteller des Lichtsignalsteuergerätes muss sicherstellen, dass die geladenen Daten auch noch zu einem späteren Zeitpunkt so zur Verfügung stehen, so dass die Checksumme korrekt gebildet werden kann und die Daten zurückgelesen werden können.

Vorschrift zur Checksummenbildung:

Die Vorschrift gilt sinngemäß für VD-Server und Lichtsignalsteuergerät, die Sortierung wird jedoch zwingend im VD-Server und bei Bedarf im Lichtsignalsteuergerät vorgenommen.

Für die Bildung der Checksumme über alle Objekte eines Blockes werden alle Objekte sortiert und in eine Folge von Bytes serialisiert. Auch bei einer optional möglichen Teilversorgung⁵ müssen die Checksummen über den gesamten Block gebildet werden. Über diese Folge wird dann ein SHA1-Digest gebildet:

- Durch die Verkettung von Member, OType und Pfad eines Objektes wird zunächst für jedes Objekt ein dem Objekt zugeordneter Bytestring, der Sortierschlüssel, gebildet. Member und OType werden dabei in Network Byte Order geschrieben.

Member	OType	Pfad
2 Byte	2 Byte	0..n Byte

Abbildung 6: Aufbau des für den Vergleich zweier Objekte genutzten Sortierschlüssel

- Mittels eines paarweisen Vergleiches dieser Sortierschlüssel beginnend beim höherwertigen Byte des Members und endend beim letzten Byte des kürzeren der Pfade beider Objekte werden die Objekte verglichen und sortiert. Die einzelnen Bytes der Schlüssel werden dabei paarweise und strikt numerisch verglichen. Sind Member, OType und der Pfad beider Objekte bis zum letzten Byte des kürzeren Pfades gleich, so ist das Objekt mit dem längeren Pfad als größer zu betrachten. In der sortierten Folge der Objekte steht nun das Objekt mit dem kleinsten Member und innerhalb der Objekte mit gleichem Member das Objekt mit dem numerisch kleinsten OType am Anfang.
- Die entstandene Folge von Objekten wird nun als Liste versorgbarer Objekte, wie für SupplyTransaction::ReadVD bzw. Transaction::AddChangeSet im Folgenden beschrieben, im BTPPL-Format serialisiert. Die in den beiden Methoden ebenfalls übertragene „Anzahl folgender VersorgbaresObjekt Elemente“ wird jedoch nicht mit in die Berechnung der Checksumme einbezogen.

⁵ Teilversorgungen sind in OCIT-O Lstg V2.0 nicht vorgesehen, die Objektmodellierung berücksichtigt jedoch diese Option auf die Zukunft.

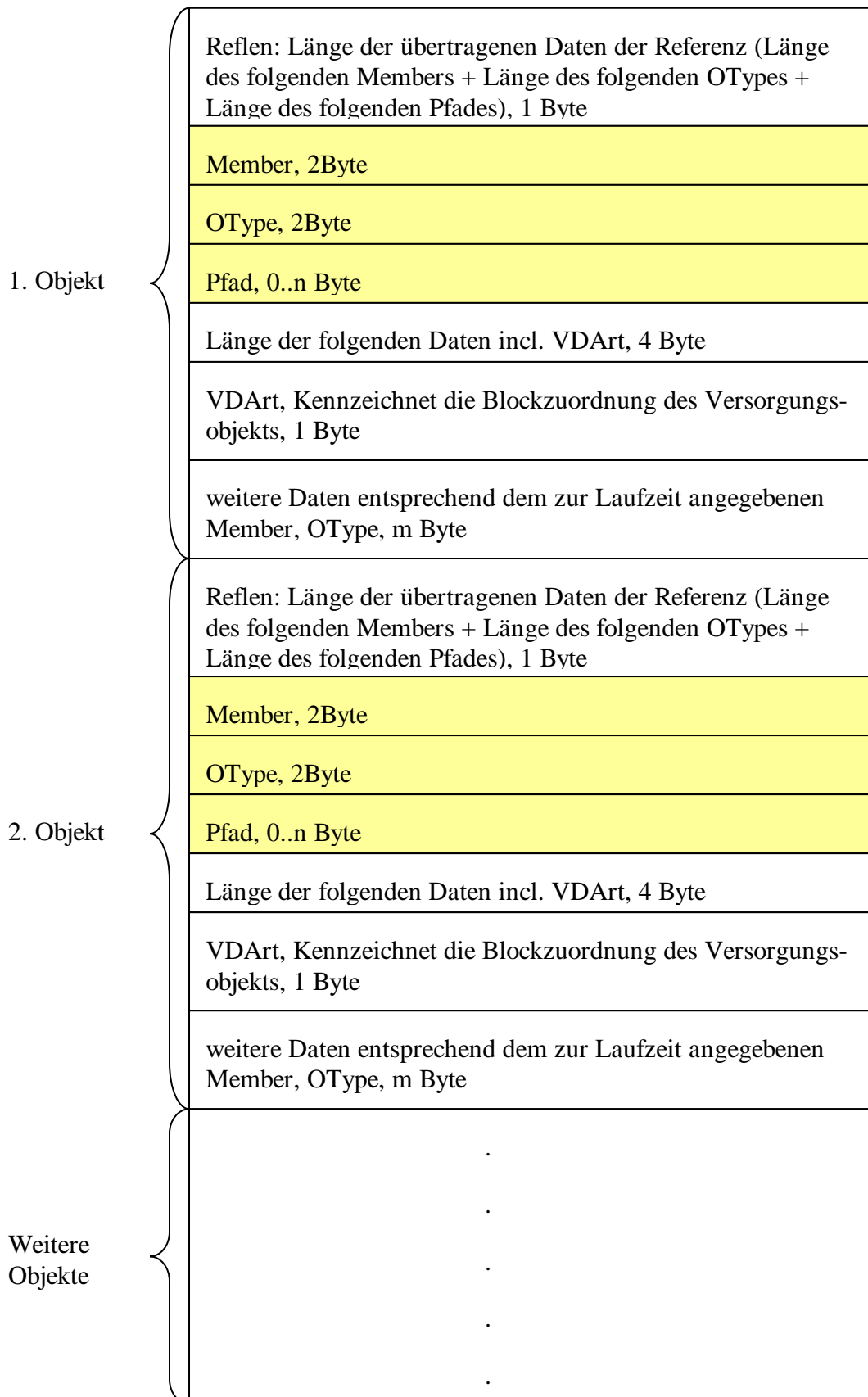


Abbildung 7: Über diese serialisierte Folge von Bytes wird ein SHA1-Digest gebildet und als Checksumme dieses Blockes verwendet.

Die Checksumme eines Blocks wird durch Aufruf der Funktion `shal_chksum(vdart)` berechnet. Die Vorschrift zur Bildung der Checksumme wird hier als C-Pseudocode dargestellt:

```
/** Die Funktion compare (cmp) wird bei der Sortierung benutzt um
    die Reihenfolge der Objekte zu definieren.
    a < b -> -1
    a == b -> 0
    a > b -> 1 */

int cmp(int a, int b) {
    if (a < b) return -1;
    else if (a == b) return 0;
    else return 1;
}

/** Funktion zum Vergleich von 2 Pfaden, um die Pfade zu sortieren. */

int path_cmp(path a, path b) {
    int member_cmp = cmp(a.member, b.member);
    if (member_cmp != 0) return member_cmp;

    int otype_cmp = cmp(a.otype, b.otype);
    if (otype_cmp != 0) return otype_cmp;

    int relknoten_cmp = cmp(a.relknoten, b.relknoten);
    if (relknoten_cmp != 0) return relknoten_cmp;

    return cmp(a.nr, b.nr);
}

/** Berechne die SHA1 Checksumme der BTPPL Daten. */
shal_chksum shal(btpplized_object p);

/** Hänge 2 BTPPL Pakete a und b aneinander */
btpplized_object concat(btpplized_object a, btpplized_coding b);

/** Berechne die Checksumme eines Versorgungsblocks vdart */
shal_chksum chkblock(int vdart) {

    // Hole alle Objekte des Blocks
    object[] o = all_objects_in_block(vdart);

    // Berechne den Pfad für alle Objekte
    path[] p = path(o);

    // Sortiere die Pfade mit der Funktion path_cmp (s.o.)
    sort(p, path_cmp);

    // coding soll die Codierung aller Pfade + Objekte
    // des Blocks enthalten
    btpplized_object coding;
    for (int i = 0; i < o.length; i++) {
        / Füge den Pfad eines Objekts zur Codierung zu
        coding = concat(coding, btppl(p[i]));
        // Füge den Aufruf von get des zum Pfad gehörenden
        // Objekts der Codierung zu.
        // Dazu sind die separat beschriebenen Bedingungen
        // für die Versorgungsobjekte zu erfüllen
        coding = concat(coding, btppl(call_method(object(p[i]), get)));
    }

    // Die eindeutige Checksumme der Versorgungsobjekte
    // eines Blocks wird hier zurückgegeben.
    return shal_chksum(coding);
}
```


}

3.2.2 Objekt Versionierung

Dieses Objekt verwaltet die Versionierungsdaten des Lichtsignalsteuergeräts (siehe Pkt.3.2.2, Tabelle 1)

3.2.2.1 Enum VDArt 1: 680

Definition der im Lichtsignalsteuergerät einzeln versionierten Versorgungsdatenarten. Diese Enumeration wird als Pfad der Versorgungsdaten Versionsinformationen (s. 3.2.2.2) verwendet.

Name	Description	Value
Grunddaten	Diese Versorgungsdatenart kennzeichnet die Grundversorgungsdaten. (Signalprogramme, verkehrstechnische Zwischen- und Versatzzeiten...)	0
Netz	Diese Versorgungsdatenart für Versorgungsdaten mit Netzbezug. Diese sind derzeit nur die Kopfdaten und die Schaltuhr (Kalender und Tagespläne).	1
VA_Steuerverfahren	Versorgungsdatenart der Steuerverfahren (XML/Binär)	2
VA_Parameter	Versorgungsdatenart der verkehrsabhängigen Programme. (Rahmenpläne, ...)	3
Gerätetechnik	Standardisierte sonstige Versorgungsdaten (Detektoren, Signalgruppen, Zuordnung zum Teilknoten, OEPNV Meldpunkte,-ketten). Proprietäre Daten Hardware Zuordnung u.a.	5
Sicherheitstechnik	Standardisierte Sicherheitsdaten, proprietäre Daten.	6

Basetyname=UBYTE maximal reservierter Wertebereich MAX=8

3.2.2.2 Objekt VDVersion

Die Meta-Daten der Versorgungsdaten, Version und Datensignatur (Checksumme und Auftraggeber) werden über den VD-Server an das Steuergerät versandt und dort abgelegt. Dazu muss pro versorgtem Block eine Instanz VDVersion in jeder Versorgungstransaktion mittels Methode AddChangeSet übertragen werden.

Diese Daten werden vom Steuergerät nur verändert, wenn eine Änderung eines Blocks der Anwenderversorgung ohne Planungstool, dh. mit herstellereigenen Mitteln erfolgt. In diesem Fall müssen im Objekt VDVersion des veränderten Blocks folgende Werte gesetzt werden:

Version: herstellereigen
Checksum: herstellereigen
Auftraggeber: herstellereigen

Das Steuergerät aktualisiert bei jeder Versorgung automatisch die zugehörig(en) LsaVersion und LsaVersionPlus Instanz(en).

Ein Get liefert wie bei allen anderen Versorgungsdatenobjekten die Werte der aktiven Versorgung

Pfad (ab Lichtsignalsteuergerät): RelKnoten(OBJECT_ID_UBYTE=UBYTE)/Art (VDArt=UBYTE)

VDVersion 1:681

VDVersion		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDArt	Zuordnung zum Versorgungsblock
	Version :string	<p>Im Fall der Versorgung über OCIT-I-VD ist dies die OCIT-I Version, die der Planer für jeden Block der Anwenderversorgung am OCIT-I VIAP vergibt.</p> <p>Im Falle der Herstellerversorgung ist das die Hersteller-Version die der Versorger für die Blöcke Geräte- und Sicherheitstechnik vergibt.</p> <p>Das Lichtsignalsteuergerät speichert die Version ohne sie zu verändern.</p>
Checksum[0..19] : ui1	<p>OCIT-I Checksumme: Für jeden Block der Anwenderversorgung werden vom Standard-Planungs- o. Versorgungswerkzeug die Checksummen erzeugt, ins Lichtsignalsteuergerät übertragen und dort gespeichert.</p> <p>Hersteller-Checksumme: Für jeden Block der Herstellerversorgung werden vom Versorgungswerkzeug die Checksummen erzeugt, ins Lichtsignalsteuergerät übertragen und dort gespeichert.</p> <p>Die Checksummen werden im Lichtsignalsteuergerät nicht verändert und können über OCIT-O ausgelesen werden.</p>	
Auftraggeber:string	<p>Für jeden Block der Anwenderversorgung werden vom Planer am OCIT-I VIAP die Namen der Auftraggeber (String 256 Zeichen) eingetragen, ins Lichtsignalsteuergerät übertragen und dort gespeichert. Diese Informationen werden im Lichtsignalsteuergerät nicht verändert und können über OCIT-O ausgelesen werden.</p>	

	OCIT_I_Session_ID	<p>Diese Session ID wird von VD-Servern nicht genutzt. Hier muss der VD-Server immer den NULLVALUE (ULONG 0xFFFFFFFF) verwenden.</p> <p>Anmerkung: Das Feldgerät kann sich hierauf nicht verlassen, da alte VD Server ggf. dieses Feld noch verwenden.</p>
--	-------------------	--

3.2.2.3 LsaVersion

Diese Versionsobjektinstanz aktualisiert das Lichtsignalsteuergerät selbstständig im Verlauf einer erfolgreichen Versorgungsdatenaktivierung.

Hinweis: Neue VD-Server sollen das Objekt LsaVersionPlus nutzen, um die Informationen abzufragen. Feldgeräte müssen die LsaVersion dennoch zur Verfügung stellen (zur Nutzung in Meldungen und aus Rückwärtskompatibilitätsgründen).

Pfad (ab Lichtsignalsteuergerät): RelKnoten(OBJECT_ID_UBYTE=UBYTE)/Art (VDArt=UBYTE)

LsaVersion 1:682

LsaVersion		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Checksum[0..19] : ui1	<p>OCIT-O Checksummen Gerät werden über die im Lichtsignalsteuergerät abgespeicherten Versorgungsdaten der Anwenderversorgung mittels des standardisierten Verfahrens gebildet.</p> <p>Hersteller-Checksummen Gerät werden über die die im Lichtsignalsteuergerät abgespeicherten Versorgungsdaten der Blöcke Geräte-und Sicherheitstechnik mittels des standardisierten oder eines proprietären Verfahrens gebildet.</p> <p>Die Checksummen können über OCIT-O ausgelesen werden.</p> <p>Gleiche Versorgungsdaten müssen gleiche Checksumme ergeben. Ist die Checksumme verschieden ist sicher, dass die zugehörigen Versorgungsdaten auch verschieden sind</p>

LsaVersion		
METHOD	Name	Beschreibung
	BuildNr	<p>Im Steuergerät wird bei jeder erfolgreichen Aktivierung eines Versorgungsblocks eine neue, den Blöcken zugeordnete Build-Nummer erzeugt. Es handelt sich dabei um einen Zähler, den das Gerät automatisch mit jeder (auch mittels lokalem herstellerspezifischem Tool vorgenommenen) Versorgung hochzählt.</p> <p>Eine Änderung der Build-Nummer signalisiert dem Versorger eine Änderung der Versorgungsdaten im Gerät. Ein Zurücksetzen ist nicht vorgesehen, der Zähler läuft einfach über. Build-Nummern können von den Versorgungswerkzeugen über OCIT-O ausgelesen werden.</p>
	Aktivierungszeitpunkt	<p>Zeitstempel Aktivierung: Der Zeitpunkt zu dem die aktuelle gültige Versorgung von einem OCIT-I VD Server aus aktiviert wurde, wird im Lichtsignalsteuergerät gespeichert und ist auslesbar</p>
	UebertragungsEndezeitpunkt	<p>Der Zeitstempel Übertragung beendet enthält die Uhrzeit beim Ende der Übertragung der Versorgungsdaten eines Blocks (=Zustandswechsel der Versorgungstransaktion nach completed).</p> <p>Er wird im Lichtsignalsteuergerät gebildet und gespeichert, er kann über OCIT-O ausgelesen werden.</p> <p>Im Fall einer Versorgung ohne Versorgungstransaktion (z.B. Herstellerspezifischen Versorgungsdatenänderung bei VDArt=Gerätetechnik und Sicherheitstechnik) ist der ÜbertragungsEndezeitpunkt gleich dem Aktivierungszeitpunkt zu setzen.</p>
	Origin	<p>Fully Qualified Domain Name (oder falls reverse lookup failed die IP-Adresse in dotted dezimal notation) von der aus die letzte erfolgreiche Versorgung dieser Art übernommen wurde.</p> <p>Im Fall der direkten lokalen Versorgung mittels nicht IP basiertem Herstellertool ist der Fully Qualified Domain Name des Geräts selbst einzutragen.</p>

3.2.2.4 LsaVersionPlus

Hinweis: Neues Objekt!

Mit diesem Objekt wird ein Fehler in der Checksummenberechnung der Versorgungsdaten behoben, der bei dem optional möglichen Vergleich OCIT-O Checksumme Server mit OCIT-O Checksumme Gerät auftritt. Dabei ändert sich die Session ID mit jedem Versorgungsvorgang. Daher ändert sich auch bei gleichbleibenden Daten bei jedem Versorgungsvorgang die Checksumme. Das neue Objekt LsaVersionPlus wurde eingeführt, um diesen Fehler zu beheben und um rückwärtskompatibel zu bleiben, das heißt, im Feld befindliche Versorgungsdaten-Server funktionieren auch mit Steuergeräten ab Ausgabe V2.0 A04.

Geräte ab Ausgabe V2.0 A04 müssen das Objekt LsaVersionPlus zusätzlich zum Objekt LsaVersion unterstützen. Das Objekt LsaVersionPlus enthält die LsaVersion und die SYSJOBID der Transaktion. Um den Fehler im System zu beseitigen, muss die SessionID von (neuem) Versorgungsdaten-Server auf NULL gesetzt werden. Die Erkennung des Ausgabestands ab A04 ist durch die Existenz des Objekts LsaVersionPlus gegeben.

Diese Versionsobjektinstanz aktualisiert das Lichtsignalsteuergerät selbstständig im Verlauf einer erfolgreichen Versorgungsdatenaktivierung. Insbesondere enthält dieses Objekt die OCIT-O-SYSJOBID der Transaction, die indentisch zur OCIT_I_Session_ID ist.

Pfad (ab Lichtsignalsteuergerät): RelKnoten (OBJECT_ID_UBYTE=UBYTE)/Art (VDArt=UBYTE)

LsaVersionPlus 1:684

LsaVersionPlus		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen

LsaVersionPlus		
METHOD	Name	Beschreibung
	Checksum[0..19] : ui1	<p>OCIT-O Checksummen Gerät werden über die im Lichtsignalsteuergerät abgespeicherten Versorgungsdaten der Anwenderversorgung mittels des standardisierten Verfahrens gebildet.</p> <p>Hersteller-Checksummen Gerät werden über die die im Lichtsignalsteuergerät abgespeicherten Versorgungsdaten der Blöcke Geräte- und Sicherheitstechnik mittels des standardisierten oder eines proprietären Verfahrens gebildet.</p> <p>Die Checksummen können über OCIT-O ausgelesen werden.</p> <p>Gleiche Versorgungsdaten müssen gleiche Checksumme ergeben. Ist die Checksumme verschieden ist sicher, dass die zugehörigen Versorgungsdaten auch verschieden sind</p>
	BuildNr	<p>Im Steuergerät wird bei jeder erfolgreichen Aktivierung eines Versorgungsblocks eine neue, den Blöcken zugeordnete Build-Nummer erzeugt. Es handelt sich dabei um einen Zähler, den das Gerät automatisch mit jeder (auch mittels lokalem herstellerspezifischem Tool vorgenommenen) Versorgung hochzählt.</p> <p>Eine Änderung der Build-Nummer signalisiert dem Versorger eine Änderung der Versorgungsdaten im Gerät. Ein Rücksetzen ist nicht vorgesehen, der Zähler läuft einfach über. Build-Nummern können von den Versorgungswerkzeugen über OCIT-O ausgelesen werden.</p>
	Aktivierungszeitpunkt	<p>Zeitstempel Aktivierung: Der Zeitpunkt zu dem die aktuelle gültige Versorgung von einem OCIT-I VD Server aus aktiviert wurde, wird im Lichtsignalsteuergerät gespeichert und ist auslesbar</p>
	UebertragungsEndezeitpunkt	<p>Der Zeitstempel Übertragung beendet enthält die Uhrzeit beim Ende der Übertragung der Versorgungsdaten eines Blocks (=Zustandswechsel der Versorgungstransaktion nach completed).</p> <p>Er wird im Lichtsignalsteuergerät gebildet und gespeichert, er kann über OCIT-O ausgelesen werden.</p> <p>Im Fall einer Versorgung ohne Versorgungstransaktion (z.B. Herstellerspezifischen Versorgungsdatenänderung bei VDArt=Gerätetechnik und Sicherheitstechnik) ist der ÜbertragungsEndezeitpunkt gleich dem Aktivierungszeitpunkt zu setzen.</p>

LsaVersionPlus		
METHOD	Name	Beschreibung
	Origin	<p>Fully Qualified Domain Name (oder falls reverse lookup failed die IP-Adresse in dotted dezimal notation) von der aus die letzte erfolgreiche Versorgung dieser Art übernommen wurde.</p> <p>Im Fall der direkten lokalen Versorgung mittels nicht IP basierendem Herstellertool ist der Fully Qualified Domain Name des Geräts selbst einzutragen.</p>
	SYSJOBID	<p>Die während einer Versorgung eines Blocks verwendete SYSJOBID der Transaction (= die OCIT_I_Session_ID die vom VD-Server vergeben wird) wird im Lichtsignalsteuergerät gespeichert und kann von den Versorgungswerkzeugen über OCIT-O ausgelesen werden.</p> <p>Bei lokalen Versorgungen wird hier die lokal erzeugte SYSJOBID eingetragen.</p>

3.2.2.5 GesamtVersion

Gesamt Buildnummer und Prüfsumme. Das Lichtsignalsteuergerät aktualisiert diese selbstständig im Verlauf einer erfolgreichen Versorgungsdatenaktivierung.

Die GesamtVersion zeigt jede Versorgungsänderung an.

Pfad (ab Lichtsignalsteuergerät): keiner, d.h. es gibt genau eine Instanz des Objekts GesamtVersion pro Lichtsignalsteuergerät.

GesamtVersion 1:683

GesamtVersion		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen

	Checksum[0..19]: ui1	<p>Die OCIT-O Checksumme Gerät gesamt wird über alle im Lichtsignalsteuergerät abgespeicherten Versorgungsdaten mittels des standardisierten oder eines proprietären Verfahrens gebildet.</p> <p>Diese Checksumme, folgt den üblichen technischen Regeln zur Bildung von Checksummen.</p> <p>Die Checksumme kann nur gelesen werden.</p> <p>Gleiche Versorgungsdaten müssen gleiche Checksumme ergeben. Ist die checksum verschieden ist sicher, dass die zugehörigen Versorgungsdaten auch verschieden sind.</p>
	BuildNr	<p>Im Steuergerät wird bei jeder erfolgreichen Aktivierung eines Versorgungsblocks eine neue, allen Blöcken zugeordnete Build-Nummer gesamt erzeugt. Es handelt sich dabei um einen Zähler, den das Gerät automatisch mit jeder (auch mittels lokalem herstellerspezifischem Tool vorgenommenen) Versorgung hochzählt.</p> <p>Eine Änderung der Build-Nummer gesamt signalisiert dem Versorger eine Änderung der Versorgungsdaten im Gerät. Ein Rücksetzen ist nicht vorgesehen, der Zähler läuft einfach über. Build-Nummern können von den Versorgungswerkzeugen über OCIT-O ausgelesen werden.</p>

3.3 Versorgungsobjekte

Folgende Versorgungsobjekte sind in OCIT-O Lstg V2.0 als „Anwenderversorgung“ standardisiert:

OType Nummern der Versorgungsobjekte, Member=1 (ODG):

OType	Name	Pfad (ab Lichtsignalsteuergerät)	VDart
669	EProgramm	RelKnoten(UBYTE)/ Nr(UBYTE)	Verkehrstechnische Grunddaten / Festzeit
670	AProgramm	RelKnoten(UBYTE)/ Nr(UBYTE)	Verkehrstechnische Grunddaten / Festzeit
666	SignalprogrammV	RelKnoten(UBYTE)/ Nr(UBYTE)	Verkehrstechnische Grunddaten / Festzeit
667	Versatzzeitenmatrix	RelKnoten(UBYTE)/ Nr(UBYTE)	Verkehrstechnische Grunddaten / Festzeit
668	VTZwischenzeitenmatrix	RelKnoten(UBYTE)/ Nr(UBYTE)	Verkehrstechnische Grunddaten / Festzeit
673	VTMinFreigabe	RelKnoten(UBYTE)/ Nr(UBYTE)	Verkehrstechnische Grunddaten / Festzeit
675	VTMinGesperrt	RelKnoten(UBYTE)/ Nr(UBYTE)	Verkehrstechnische Grunddaten / Festzeit
650	Kopfdaten	RelKnoten(UBYTE)	Daten mit Netzbezug
660	Tagesplan	RelKnoten(UBYTE)/Nr(UBYTE)	Daten mit Netzbezug
661	Wochenplan	RelKnoten(UBYTE)/Nr(UBYTE)	Daten mit Netzbezug
662	SondertagJaehrlich	RelKnoten(UBYTE)/Nr(UBYTE)	Daten mit Netzbezug
663	SondertagAufzaehlung	RelKnoten(UBYTE)/Nr(UBYTE)	Daten mit Netzbezug
664	Zeitbereich	RelKnoten(UBYTE)/Nr(UBYTE)	Daten mit Netzbezug
672	BinaerVASteuerverfahren	RelKnoten(UBYTE))/ Kennung (STRING)	VA-Steuerverfahren
676	BinaerVAParameter	RelKnoten(UBYTE))/-Kennung (STRING)	VA-Parameter

3.3.1 Objekt VersorgbaresObjekt

VersorgbaresObjekt ist die BASEDOMAIN für alle versorgbaren Objekte.

VersorgbaresObjekt (1:648)

VersorgbaresObjekt		
METHOD	Name	Beschreibung

16	GetVDArt	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDArt	Kennzeichnet die Blockzuordnung des Versorgungsobjekts.

Folgende Objekte sind von „VersorgbaresObjekt“ abgeleitet:

EProgramm
 AProgramm
 SignalprogrammV
 Versatzzeitenmatrix
 VTZwischenzeitenmatrix
 VTMinFreigabe
 VTMinGesperrt
 Kopfdaten
 Tagesplan
 Wochenplan
 Sondertag
 Zeitbereich
 BinaerVASteuerverfahren
 BinaerVAParameter
 VDVersion

3.3.2 Block 1: Verkehrstechnische Grunddaten / Festzeit

3.3.2.1 Objekt Eprogramm

In diesem Objekt werden Einschaltprogramme gespeichert.

Während des Einschaltprogramms gelten die Zwischenzeiten des Zielprogramms.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

Wie in der folgenden Methodenbeschreibung eingetragen.

- EAZeile Elemente sortiert aufsteigend nach Signalgruppennummer.
- Schaltzeit Elemente sortiert aufsteigend nach Schaltzeitpunkt.

Eprogramm (1:669)

Eprogramm			
METHOD	Name	Beschreibung	
0	Get		
	Ausgabeparameter		
	RetCode	OK: folgende Parameter richtig gelesen	
	VDArt	Zuordnung zum Versorgungsblock.	
	Bezeichnung: bezeichnungType	Signalprogrammname, (ist nur für Rücklesen der Versorgung in OTEC Format nötig).	
	Dauer	Dauer des Einschaltprogramms	
	Signalsicherungszeitpunkt	Zeitpunkt, an dem die Signalsicherung eingeschaltet wird.	
	Anzahl	Anzahl folgender EAZeile Elemente	
	EAZeile[1..254]		
		Signalgruppe.Nr	Nummer der zu schaltenden Signalgruppe
		Anzahl	Anzahl folgender SchaltzeitTyp Elemente
	SchaltzeitTyp[1...254]		
		Schaltzeit	Schaltung der Signalgruppe in ein neues Signalbild. Schaltzeit einer Signalgruppe. Zum gleichen Schaltzeitpunkt darf nur eine Zielfarbe angegeben werden.
		Signalbild	Zu schaltendes Signalbild

Hinweis: Der Zustand der Signalgruppen vor dem Einschalten wird bis zum ersten Schaltbefehl der jeweiligen Signalgruppe des Einschaltprogramms beibehalten!

3.3.2.2 Objekt AProgramm

In diesem Objekt werden Ausschaltprogramme gespeichert.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

Wie in der folgenden Methodenbeschreibung eingetragen.

- EAZeile Elemente sortiert aufsteigend nach Signalgruppennummer.
- Schaltzeit Elemente sortiert aufsteigend nach Schaltzeitpunkt.

AProgramm (1:670)

AProgramm			
METHOD	Name	Beschreibung	
0	Get		
	Ausgabeparameter		
	RetCode	OK: folgende Parameter richtig gelesen	
	VDart	Zuordnung zum Versorgungsblock.	
	Bezeichnung: bezeichnungType	Signalprogrammname, (ist nur für Rücklesen der Versorgung in OTEC Format nötig).	
	Dauer	Dauer des Ausschaltprogramms	
	Signalsicherungszeitpunkt	Zeitpunkt, an dem die Signalsicherung ausgeschaltet wird.	
	Anzahl	Anzahl folgender EAZeile Elemente	
	EAZeile[1..254]		
		Signalgruppe.Nr	Nummer der zu schaltenden Signalgruppe
		Anzahl	Anzahl folgender SchaltzeitTyp Elemente
	SchaltzeitTyp[1...254]		
		Schaltzeit	Schaltung der Signalgruppe in ein neues Signalbild. Schaltzeit einer Signalgruppe. Zum gleichen Schaltzeitpunkt darf nur eine Zielfarbe angegeben werden.
		Signalbild	Zu schaltendes Signalbild

Hinweis: Der Zustand der Signalgruppen im Ausschaltzeitpunkt wird bis zum ersten Schaltbefehl der jeweiligen Signalgruppe des Ausschaltprogramms beibehalten!

3.3.2.3 Objekt SignalprogrammV

In diesem Objekt werden Signalprogrammdaten gespeichert.

Ein- und Ausschaltprogramme werden in gesonderten Objekten verwaltet.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

Wie in der folgenden Methodenbeschreibung eingetragen.

- SPZeile Elemente sortiert aufsteigend nach Signalgruppennummer.
- Referenzuebergang Elemente sortiert nach aufsteigend Startbild, für die gleiche Starbilder sortiert aufsteigend nach Zielbild.
- Schaltzeit Elemente sortiert aufsteigend nach Schaltzeitpunkt.

Hinweis: Bei Blockversorgung müssen immer alle im Lichtsignalsteuergerät vorhandenen Signalprogramme versorgt werden. Es können keine neuen Signalprogramme hinzugefügt oder bestehende gelöscht werden.

SignalprogrammV (1:666)

SignalprogrammV		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDart	Zuordnung zum Versorgungsblock
	Bezeichnung: BezeichnungType	Signalprogrammname (ist nur für Rücklesen der Versorgung in OTEC Format nötig).
	ZWZMatrix.Nr	Nummer der verwendeten Zwischenzeitmatrix. Wenn die Sicherheits-ZWZ-Matrix verwendet wird, steht an der Stelle 0
	VZMatrix.Nr[0 ... 2]: ui1	Nummer der verwendeten Versatzzeitenmatrix: 0 = BB 1 = EE 2 = BE Ist eine Matrix nicht definiert, steht an der entsprechenden Position ein NULLVALUE (0xFF). Die Versorgung von nicht unterstützten VZMatrix-Arten darf vom Gerät zurückgewiesen werden.
	VTMinFreigabe.Nr	Nummer der verwendeten Mindestfreigabezeitenliste. Wenn die Sicherheits-Mindestfreigabezeiten verwendet werden, steht an der Stelle 0.

SignalprogrammV		
METHOD	Name	Beschreibung
	VTMinGesperrt.Nr	Nummer der verwendeten Mindestgesperrtzeitenliste. Wenn die Sicherheits-Mindest- Gesperrtzeiten verwendet werden, steht an der Stelle 0.
	TU	Umlaufzeit des Programms
	EP	Einschaltpunkt, sofern definiert, sonst NULLVALUE
	AP	Ausschaltpunkt, sofern definiert, sonst NULLVALUE
	UP	Kopfdaten Umschaltpunkt (in Siemens-Terminologie GSP), sonst NULLVALUE.
	SY_Vor	Vorrastpunkt, falls vorhanden, sonst NULLVALUE
	SY_Haupt	Hauptrastpunkt, falls vorhanden, sonst NULLVALUE.
	SY_MaxDauer	Maximale Dauer, für die synchronisiert wird falls vorhanden, sonst NULLVALUE.
	SignalzeitenVersatz	Versatzzeit bezogen auf die Versatzzeit-Berechnungsvorschrift für die Synchronisierung des Gerätes. Der Wert kann auf Geräten mit unterschiedlichen Versatz-Berechnungen zu verschiedenen Zeiten führen
	EProgramm.Nr	Zugehöriges Einschaltprogramm Der OCIT-I Versorgungsdatenserver muss prüfen, ob das Einschaltprogramm vorhanden ist.
	AProgramm.Nr	Zugehöriges Ausschaltprogramm Der OCIT-I Versorgungsdatenserver muss prüfen, ob das Ausschaltprogramm vorhanden ist.
	Anzahl	Anzahl folgender SPZeile Elemente
	Signalgruppe.Nr	Nummer der zu schaltenden Signalgruppe
	Anzahl	Anzahl folgender Referenzen auf Zusatzübergänge [0 .. 2]
	ReferenzUebergang	Verweis auf Zusatzübergänge die in den Signalgruppensdaten definiert sind.
	Anzahl	Anzahl folgender SchaltzeitType Elemente
	Schaltungen oder Dauersignalbild [1 ... 10]	
	Schaltzeitpunkt	Schaltzeit der Signalgruppe (NULLVALUE = Dauersignalbild)
	Signalbild	Zu schaltendes Signalbild

Hinweis: SY_Vor und SY_Haupt sind Synchronisierungspunkte zum Dehnen und Stauchen eines Signalplans. Bei Steuerverfahren, die diese Methode verwenden, läuft der TX bis zum SY_Vor, springt direkt auf SY_Haupt und wartet dort auf die Synchronisierung. Achtung: Schaltbefehle zwischen SY_Vor und SY_Haupt sind unzulässig und werden nicht ausgeführt.

3.3.2.3.1 OCIT-O Referenzen auf Zusatzübergänge

In OCIT-O werden Zusatzübergänge über die im OCIT-I Format vorliegenden Namen der Zusatzübergänge referenziert.

In OCIT-I sind die Zusatzübergänge in den Signalgruppendaten der „OCIT-I_VD-DM-LSA.xsd“ definiert. Die im Signalprogramm gültigen Zusatzübergänge werden in den Signalplanzeilen für jede Signalgruppe referenziert. Der Name ist ein String beliebiger Länge, wobei die Namensgebung standardisiert ist, um das Format leichter lesbar zu machen. Der Name setzt sich aus dem Bitcode des Start-Signalbildes, den Namen aller Übergangs-Signalbilder inkl. Dauer und dem End-Signalbild zusammen.

Um in OCIT-O auf eine kompakte Darstellung der Referenz auf einen Zusatzübergang zu kommen wird der Name wie in Pkt. 3.3.2.3.1.1 beschrieben kodiert. Aufgrund der Darstellung der Namen in der OCIT-I_VD-DM-LSA.xsd, die über Pattern vorgegeben ist, ist eine einfache Umwandlung möglich. Auch eine Zuordnung der Referenzen auf die im Gerät vorhanden hinterlegten Zusatzübergänge ist möglich.

Es wird festgelegt das bei OCIT-O maximal 3 Übergangselemente pro Übergang möglich sind.

Hinweise: Die Versorgung kann abgelehnt werden, wenn die übergebenen Zusatzübergänge nicht in der Signalsicherung vorhanden sind. Bei der Verwendung von Zusatzübergängen sind gerätespezifische Grenzen besonders kritisch, da die resultierenden Kombinationen die Möglichkeiten der Geräte überschreiten können.

3.3.2.3.1.1 Kodierung der Referenz auf einen Zusatzübergang

Kodierung als ui8 :

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Startbild: ui1	Länge 1. Übergangselement: ui1	Signalbild 1. Übergangselement: ui1	Länge 2. Übergangselement: ui1	Signalbild 2. Übergangselement: ui1	Länge 3. Übergangselement: ui1	Signalbild 3. Übergangselement: ui1	Zielbild

Nicht verwendete Übergangselemente werden mit der Länge (0x0) und Signalbild dunkel (0x0) kodiert.

Beispiel : Übergang „rot-1srotgelb-gruen“

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x03 (rot)	0x0a (10 x 100 ms)	0x0f (rtge)	0x0	0x0	0x0	0x0	0x30 (gruen)

Achtung: Bei dieser Kodierung handelt es sich ausschließlich um eine Referenz (mittels Namen) auf einen Zusatzübergang. Es **handelt sich nicht um die realen Daten** des Zusatzübergangs.

3.3.2.4 Objekt Versatzzeitenmatrix

In diesem Objekt werden die Versatzzeitenmatrizen gespeichert.

Im Steuergerät sind bis zu neun verschiedene Versatzzeitenmatrizen vorhanden, von denen bis zu drei in der Signalprogrammversorgung ausgewählt werden.

Es darf gleichzeitig nur eine Versatzzeitenmatrix gleicher Art angewählt werden.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

Wie in der folgenden Methodenbeschreibung eingetragen.

- Zeilen aufsteigend sortiert nach SGrBasis, dann nach SGrAbhaengig, dann nach Operator.

Hinweis: Die Versatzzeiten können vom VA-Verfahren verwendet werden. Der Umgang mit den Versatzzeiten ist nicht standardisiert.

Versatzzeitenmatrix (1:667)

Versatzzeitenmatrix		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDart	Zuordnung zum Versorgungsblock.
	Bezeichnung: bezeichnungType	Name der Versatzzeitenmatrix, (ist nur für Rücklesen der Versorgung in OTEC Format nötig).
	Art	Versatzbedingungen zwischen- Beginn → Beginn- Ende → Ende-Beginn → Ende.
	Anzahl	Anzahl folgender VersatzZeitEintrag Elemente
	VersatzZeitEintrag [0..9999]	Eintrag in der Versatzzeitmatrix. Der Eintrag bezieht sich immer auf das Verhältnis der Abhängigen Signalgruppe zur Basissignalgruppe. Beispiel: Bei Art = "BeginnBeginn" SGrBasis = "Bas", SGrAbhaengig = "Abh", Wert = 12 und Operator = "gleich" muss der Beginn der Freigabezeit der Signalgruppe "Abh" 12 Sekunden hinter der Signalgruppe "Bas" liegen.
	SGrBasis.Nr	Nummer der Basis-Signalgruppe
	SGrAbhaengig.Nr	Nummer der abhängigen Signalgruppe
Wert	Zeitwert (kann negativ sein)	
Operator		

Art (VersatzzeitArt):

Wert	Bedeutung	Name
1	diese Art kennzeichnet Versatzzeiten von Beginn zu Beginn	BeginnBeginn
2	diese Art kennzeichnet Versatzzeiten von Ende zu Ende	EndeEnde

3	diese Art kennzeichnet Versatzzeiten von Beginn zu Ende	BeginnEnde
---	---	------------

Hinweis: Es ist nicht verbindlich festgelegt, dass ein OCIT-O Lichtsignalsteuergerät alle aufgeführten Versatzzeitenarten beherrschen muss.

Operator (VersatzzeitOperator):

Wert	Bedeutung	Name
1	größergleich (der Zeitwert der abhängigen Signalgruppe muss größer oder gleich dem Zeitwert der maßgebenden Signalgruppe (Basissignalgruppe) sein)	ge
2	kleinergleich (der Zeitwert der abhängigen Signalgruppe muss kleiner oder gleich dem Zeitwert der maßgebenden Signalgruppe (Basissignalgruppe) sein)	le
3	gleich (der Zeitwert der abhängigen Signalgruppe muss gleich dem Zeitwert der maßgebenden Signalgruppe (Basissignalgruppe) sein)	eq

Es gilt die mathematische Sichtweise z. B. $-6 \text{ le } -5$.

3.3.2.4.1 Versatzvarianten (Beispiele)

Die Einhaltung von Versatzbedingungen ist gegenüber der Einhaltung von Zwischen- und Mindestzeiten nachrangig.

Stellt sich bei Betrachtung der Versätze heraus, dass Versatzzeiten nicht eingehalten werden können, wird der Schaltvorgang entsprechend verschoben.

3.3.2.4.1.1 Feste Versätze

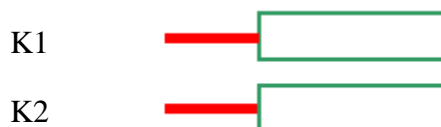
Feste Versätze werden bei OCIT mit dem ‚eq‘ - Operator (gleich) versorgt

Bei festen Versätzen sind beide Signalgruppen gegenseitig von einander abhängig, d.h. es ist keine Zuordnung zwischen maßgebender und abhängiger Signalgruppe möglich. Die Signalgruppen müssen gemäß dem festgelegten Versatz schalten.

Fester Beginn-Versatz (BeginnBeginn):

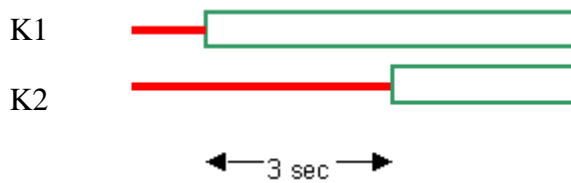
1) Beide schalten zum gleichen Zeitpunkt:

	K1	K2
K1		0
K2	0	



2) K2 schaltet genau 3 Sekunden nach K1:

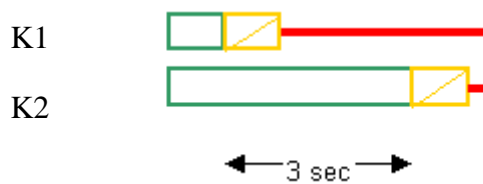
	K1	K2
K1		3
K2	-3	



Fester Ende-Versatz (EndeEnde):

K2 schaltet genau 3 Sekunden nach K1.

	K1	K2
K1		3
K2	-3	



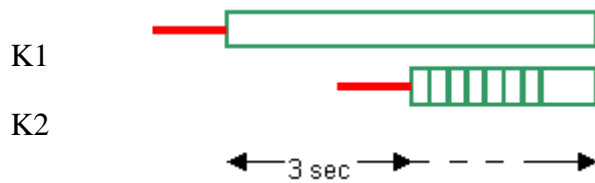
3.3.2.4.1.2 Variable Versätze

Variable Versätze werden bei OCIT mit den ‚ge‘ - (größer gleich) oder ‚le‘ - Operator (kleiner gleich) versorgt.

Variabler Beginn-Versatz (BeginnBeginn)

1) K2 schaltet 3 Sekunden oder mehr nach K1.

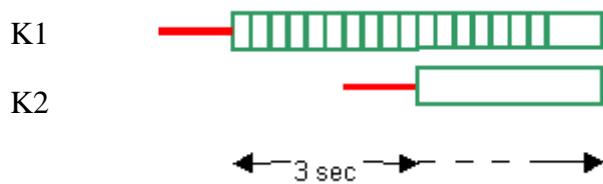
	K1	K2
K1		3
K2		



Hier ist K1 maßgebend, K2 abhängig. Dies wird bei OCIT mit dem ‚ge‘ - Operator (größer gleich) versorgt.

2) K1 schaltet 3 Sekunden oder weniger vor K2 (eventl. auch nach K2).

	K1	K2
K1		
K2	-3	

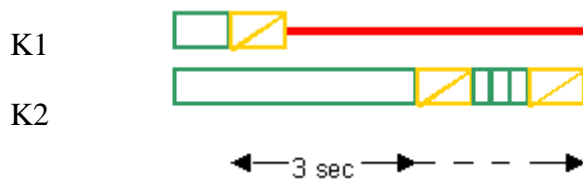


Hier ist K2 maßgebend, K1 abhängig. Dies wird bei OCIT mit dem ‚le‘ - Operator (kleiner gleich) versorgt.

Variabler Ende-Versatz (EndeEnde)

1) K2 schaltet 3 oder mehr Sekunden nach K1.

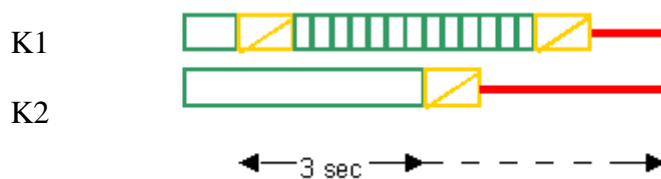
	K1	K2
K1		3
K2		



Hier ist K1 maßgebend, K2 abhängig. Dies wird bei OCIT mit dem ‚ge‘ - Operator (größer gleich) versorgt.

2) K2 schaltet weniger als 3 Sekunden vor K1.

	K1	K2
K1		
K2	-3	



Hier ist K2 maßgebend, K1 abhängig. Dies wird bei OCIT mit dem ‚le‘ - Operator (kleiner gleich) versorgt.

3.3.2.5 Objekt VTZwischenzeitenmatrix

In diesem Objekt werden die verkehrstechnischen Zwischenzeitenmatrizen für besondere Signalprogramme (z.B. Schlechtwetter) gespeichert. Verkehrstechnische Zwischenzeitenmatrizen haben die Nummern 1..3. Alle Zeitwerte der VTZwischenzeitenmatrix müssen grösser oder gleich den Zeitwerten der sicherheitsrelevanten Zwischenzeitenmatrix sein. Es kann jeweils nur eine VTZwischenzeitenmatrix aktiv sein.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

Wie in der folgenden Methodenbeschreibung eingetragen.

- ZwischenZeitEintrag aufsteigend sortiert nach Raeumer, dann nach Einfahrer.

VTZwischenzeitmatrix (1:668)

VTZwischenzeitmatrix		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDart	Zuordnung zum Versorgungsblock.
	Bezeichnung: bezeichnungType	Signalprogrammname (ist nur für Rücklesen der Versorgung in OTEC Format nötig).
	Anzahl	Anzahl folgender VersatzZeitEintrag Elemente
	ZwischenZeitEintrag [0..9999]	Eintrag in der VTZwischenzeitenmatrix
	Raeumer.Nr	Nummer der Räumenden Signalgruppe
	Einfahrer.Nr	Nummer der Einfahrenden Signalgruppe
Wert	Zeitwert	

3.3.2.6 Objekt VTMinFreigabe

In diesem Objekt werden die verkehrstechnischen Mindestfreigabezeiten für besondere Signalprogramme (z.B. Schlechtwetter) gespeichert. Die Listen der verkehrstechnische Mindestfreigabezeiten haben die Nummern 1..3. Alle Zeitwerte der VTMinFreigabe müssen grösser oder gleich den Zeitwerten der sicherheitsrelevanten Mindestfreigabezeiten sein. Es kann jeweils nur eine Mindestfreigabezeitenliste aktiv sein.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

Wie in der folgenden Methodenbeschreibung eingetragen.

- Zeiten aufsteigend sortiert nach Signalgruppennummer.

VTMinFreigabe (1:673)

VTMinFreigabe		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDArt	Zuordnung zum Versorgungsblock.
	Bezeichnung: bezeichnungType	Name der Mindesfreigabezeitliste, (ist nur für Rücklesen der Versorgung in OTEC Format nötig).
	Anzahl	Anzahl folgender Mindestzeiten
	zeit[0..254]	
		SG.Nr
	Wert	Zeitwert

3.3.2.7 Objekt VTMinGesperrt

In diesem Objekt werden die verkehrstechnischen Mindestsperrzeiten für besondere Signalprogramme (z.B. Schlechtwetter) gespeichert. Die Listen der verkehrstechnische Mindestsperrzeiten haben die Nummern 1..3. Alle Zeitwerte der VTMinGesperrt müssen grösser oder gleich den Zeitwerten der sicherheitsrelevanten Mindestsperrzeiten sein. Es kann jeweils nur eine Mindestsperrzeitenliste aktiv sein.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

Wie in der folgenden Methodenbeschreibung eingetragen.

- Zeiten aufsteigend sortiert nach Signalgruppennummer.

VTMinGesperrt (1:675)

VTMinGesperrt		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDArt	Zuordnung zum Versorgungsblock.
	Bezeichnung: bezeichnungType	Name der Mindesgesperrtzeitliste, (ist nur für Rücklesen der Versorgung in OTEC Format nötig).
	Anzahl	Anzahl folgender Mindestzeiten

VTMinGesperrt		
METHOD	Name	Beschreibung
	zeit[0..254]	
	SG.Nr	Nummer der Signalgruppe
	Wert	Zeitwert

3.3.3 Block 2: Daten mit Netzbezug

3.3.3.1 Objekt Kopfdaten

Dieses Objekt speichert die Grunddaten des Lichtsignalsteuergeräts. Diese Daten haben nur informativen Charakter für Bediener und werden nicht in die Ablaufsteuerung des Lichtsignalsteuergeräts übernommen.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

- Wie in der folgenden Methodenbeschreibung eingetragen.

Kopfdaten (1:650)

Kopfdaten		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDart	Zuordnung zum Versorgungsblock.
	Kurzbezeichnung: kurzStringType	Kurzbezeichnung der Anlage z.B. K123
	Name: langStringType	Langname des Knotens z.B. Meierstraße / Müllerstraße
	UnitID	Logische Adressierung des Knotenpunktes aus der Sicht OCIT-I
	SystemNr	Amts- bzw. Bezirkskennung
	SubSystemNr.	Stadt, falls in OTEC nicht gesetzt, mit NULLVALUE belegt
	UnitNr	Eindeutige Nummer des Knotenpunktes im Steuerungsgebiet (1 bis 4294967295)
	Bemerkung: langStringType	kundenspezifische Bemerkungen

3.3.3.2 Schaltuhr

Die Objekte Tagesplan, Wochenplan, SondertagJaehrlich, SondertagAufzaehlung und Zeitbereich beinhalten die Daten der Schaltuhr (Jahresuhr, Jahresautomatik, JAUT) des Lichtsignalsteuergeräts.

3.3.3.2.1 Projektspezifische Modifikationen der Schaltuhr

Insgesamt sind 16 Modifikationen verfügbar. 3 Modifikationen (ModVA, ModOepnv, ModVAIndividualverkehrEinAus) sind bereits belegt, 13 Modifikationen sind für die projektspezifische Nutzung über die Schaltuhr frei. Diese projektspezifischen Modifikationen haben jeweils eine Nummer, die von 0 bis 254 laufen kann. Für 8 davon werden Empfehlungen zur Belegung gegeben. Sie umfassen die Nummer und den Namen der Modifikation. Jede projektspezifische Modifikation kann Ein oder Aus geschaltet werden. Der Name der projektspezifischen Modifikation ist versorgbar (Herstellerversorgung) und auslesbar.

Da die Modifikationen nicht nur von der Schaltuhr, sondern auch von der Zentrale aus geschaltet werden können, wird festgelegt dass die Zentrale Priorität hat.

Hinweis: Die mit den Modifikationen geschalteten Anwendungen sind nicht standardisiert und müssen projektspezifisch vereinbart werden!

Projektspezifische Modifikationen der Schaltuhr (Empfehlungen zur Belegung)		
Nr.	Name	Bemerkung
0	Blindenakustik Orientierungston	
1	Blindenakustik Freigabeton	
2	Detektorüberwachung	Falls mehrere Überwachungszeiten, z.B. Morgenspitze, Nachmittagsspitze, Normalverkehr, Schwachverkehr notwendig sind, müssen dazu noch freie projektspezifische Modifikationen benutzt werden.
3	Digitaler Ausgang A	Angabe der Kanalnummern und DigAusgang::Get wird derzeit nicht unterstützt.
4	Digitaler Ausgang B	
5	Digitaler Ausgang C	
6	Digitaler Ausgang D	
7	LSA-Standby	Zum Beispiel Anlage Dunkel oder alles Rot.

3.3.3.2.2 Objekt Tagesplan

In diesem Objekt werden die Daten der Tagespläne gespeichert.

Es ist mindestens ein Standard-Tagesplan vorhanden, der immer dann ausgeführt wird, wenn keine andere Regelung greift. Der Standard-Tagesplan hat die Nr 1.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden: Wie in der folgenden Methodenbeschreibung eingetragen.

- Befehle mit aufsteigender Uhrzeit.
- Teilknoten Elemente sortiert aufsteigend nach Teilknotennummer
- Projektspezifische Modifikation Elemente sortiert nach Nummer der Modifikation

Tagesplan (1:660)

Tagesplan		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDart	Zuordnung zum Versorgungsblock.
	BezeichnungKurz:	Bezeichnung des Tagesplans.
	bezeichnungType	Die Bezeichnung innerhalb der Tagespläne muss eindeutig sein
	Anzahl	Anzahl folgender Jaut-Befehl Elemente
	Befehl[0 ... 254] Die Befehle müssen mit aufsteigender Uhrzeit sortiert sein	
	Uhrzeit	Uhrzeit, an dem der Befehl begonnen wird. Zu einem Zeitpunkt kann nur ein Befehl begonnen werden, da keine Nullvalues zugelassen sind.. Als Uhrzeit werden die Sekunden seit Mitternacht lokaler Zeit angegeben. Bei Winter-Sommerzeitumstellung wird der letzte übersprungene Schaltzeitpunkt nachgeholt.
	Programmwunsch	Programmwunsch
	KnotenEinAus	Gesamtknoten Ein/Aus
	ModVA	Verkehrsabhängigkeit Ein/Aus
	ModOepnv	OEPNV Bevorzugung
	ModVAIndividualverkehrEinAus	Dieser Teilbefehl schaltet die Verkehrsabhängige Beeinflussung des Signalprogramms durch Individualverkehr Ein/Aus.
	Anzahl	Anzahl folgender Teilknotenzustände. Die Anzahl der TK muss mit den tatsächlich vorhandenen übereinstimmen!
	TkZustand[0...3]	
TeilKnotenNr.	Teilknotennummer	
SollZustand	Teilknoten Ein/Aus	
Modifikation[0 ... 12]	Projektspezifische Modifikationen (alle 13)	

Tagesplan			
METHOD	Name		Beschreibung
		Nr.ui1	Nummer der Modifikation
		Wert	Wert der Modifikation. Nicht genutzte Modifikationen werden mit Wert „Aus“ belegt!

Hinweis: Jeder Tagesplan muss mindestens einen Eintrag haben. Der Schaltwunsch muss vollständig sein, NULLVALUES und Zustand „keiner“ sind generell nicht erlaubt!

3.3.3.2.3 Objekt Wochenplan

In diesem Objekt werden die Daten der Wochenpläne gespeichert.

Es ist mindestens ein Standard-Wochenplan vorhanden, der immer dann ausgeführt wird, wenn keine andere Regelung greift. Der Standard Wochenplan hat die Nr 1.

Festlegung zur Reihenfolge in der die Versorgungsdaten werden sortiert und gelesen werden:

- Wie in der folgenden Methodenbeschreibung eingetragen.

Wochenplan (1:661)

Wochenplan			
METHOD	Name		Beschreibung
0	Get		
	Ausgabeparameter		
	RetCode		OK: folgende Parameter richtig gelesen
	VDArt		Zuordnung zum Versorgungsblock.
	BezeichnungKurz:		Bezeichnung des Wochenplans.
	bezeichnungType		Die Bezeichnung innerhalb der Wochenpläne muss eindeutig sein. (ist nur für Rücklesen der Versorgung in OTEC Format nötig)
	Mo		Nummer des Tagesplans, der am Montag ausgeführt wird
	Di		Nummer des Tagesplans, der am Dienstag ausgeführt wird
	Mi		Nummer des Tagesplans, der am Mittwoch ausgeführt wird
	Do		Nummer des Tagesplans, der am Donnerstag ausgeführt wird
	Fr		Nummer des Tagesplans, der am Freitag ausgeführt wird
	Sa		Nummer des Tagesplans, der am Samstag ausgeführt wird
	So		Nummer des Tagesplans, der am Sonntag ausgeführt wird

Hinweis: NULLVALUES sind nicht erlaubt!

3.3.3.2.4 Objekt SondertagJaehrlich

In diesem Objekt werden die Daten aller jährlich wiederkehrenden Feier- oder Sondertage gespeichert.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

- Wie in der folgenden Methodenbeschreibung eingetragen.

Festlegung zur Priorität:

- Größerer Zahlen bedeuten höhere Gewichtung (Priorität):
- 0 Standardgewichtung von normalen Tagen
 1 Standardgewichtung von Zeitbereichstagen (Ferien)
 2 Standardgewichtung von SondertagJaehrlich (Feiertagen)
 3 Standardgewichtung von SondertagAufzaehlung
 4..9 Hochpriore Sondertage für Spezialfälle

Bei gleichen Prioritäten gilt die Reihenfolge Zeitbereich, SondertagJaehrlich und Sondertag Aufzaehlung (höchste Priorität).

SondertagJaehrlich (1:662)

SondertagJaehrlich		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDArt	Zuordnung zum Versorgungsblock.
	Name: bezeichnungType	Name des Feiertags (ist nur für Rücklesen der Versorgung in OTEC Format nötig).
	Tagesplan	Referenz auf den Tagesplan, der an diesem Tag verwendet wird.
	Prioritaet	Wertebereich 1-9, da 0 für normale Tage reserviert (siehe oben)
	Datum	Tagescodierung siehe Feiertag. Achtung! Im Fall von Sondertagen werden hier auch nicht im Enum definierte Werte eingetragen.

Die beweglichen und festen Feiertage werden im Tagescode wie folgt codiert:

Feiertage mit festem Datum:

Feiertage mit festem Datum werden über den Zahlenbereich 0-365 abgedeckt.

Das Datum des Feiertags wird als Tag im Jahr abgelegt (Beginnend ab 0 für 1.1.).

Bei der Bestimmung des Tags wird immer ein Schaltjahr angenommen (Februar=29 Tage, Jahr = 366 Tage).

Osterabhängige Feiertage:

Bei Feiertagen, die vom Ostersonntag abhängig sind, wird zu der Differenz zum Ostersonntag der Wert 500 addiert.

Wenig schwankende Feiertage am gleichen Wochentag:

Wenig schwankende Feiertage , die nicht von Ostern abhängig sind, sind Feiertage, die immer am selben Wochentag stattfinden.

Es wird der erste Tag angegeben, an dem der Feiertag stattfinden kann (beginnend ab 0 für 1.1.). Danach wird je nach Wochentag ein Offset aufaddiert:

Mo=1000

Di=2000

Mi=3000

Do=4000

Fr=5000

Sa=6000

So=7000

Bei der Bestimmung des Tags wird immer ein Schaltjahr angenommen (Februar = 29 Tage, Jahr = 366 Tage).

Beispiele für Feiertage:

Name	Berechnung	Wert
Neujahr	fest 1.1.	0
Epiphania	fest 6.1.	5
Tag der Arbeit	fest 1.5.	121
Tag der deutschen Einheit	fest 3.10.	276
Reformationsfest	fest 31.10.	304
Allerheiligen	fest 1.11.	305
Weihnachtsfeiertag1	fest 25.12.	359
Weihnachtsfeiertag2	fest 26.12.	360
Fastnacht	OSTERN-47	453
Karfreitag	OSTERN-2	498
Ostersonntag	OSTERN	500
Ostermontag	OSTERN+1	501
Muttertag	Sonntag im Bereich 8.-14.Mai	7128
Christi Himmelfahrt	OSTERN+39	539
Pfingstsonntag	OSTERN+49	549
Pfingstmontag	OSTERN+50	550
Fronleichnam	OSTERN+60	560
Maria Himmelfahrt	fest am 15.8.	227
Buß und Betttag	Mittwoch im Bereich 16.-22. November	3320

Maria Empfängnis	fest 8.12.	342
------------------	------------	-----

3.3.3.2.5 Objekt SondertagAufzaehlung

In diesem Objekt werden die einmaligen Feier- oder Sondertage mit vorgegebenem Datum gespeichert.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

Wie in der folgenden Methodenbeschreibung eingetragen.

Festlegung zur Priorität:

- Größerer Zahlen bedeuten höhere Gewichtung (Priorität):
- 0 Standardgewichtung von normalen Tagen
- 1 Standardgewichtung von Zeitbereichstagen (Ferien)
- 2 Standardgewichtung von Feiertagen
- 3 Standardgewichtung von Sondertagen
- 4..9 Hochpriorie Sondertage für Spezialfälle

Bei gleichen Prioritäten gilt die Reihenfolge Zeitbereich, SondertagJaehrlich und Sondertag Aufzaehlung (höchste Priorität).

SondertagAufzaehlung (1:663)

SondertagAufzaehlung		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDart	Zuordnung zum Versorgungsblock.
	Name: bezeichnungType	Name des Feiertags. (ist nur für Rücklesen der Versorgung in OTEC Format nötig)
	Tagesplan.	Referenz auf den Tagesplan, der an diesem Tag verwendet wird.
	Prioritaet	Wertebereich 1-9, da 0 für normale Tage reserviert (siehe oben)
	Tag	Tag des Sondertags
	Monat	Monat des Sondertags
	Jahr	Jahr des Sondertags

3.3.3.2.6 Objekt Zeitbereich

In diesem Objekt werden die Schaltuhr-Zeitbereiche gespeichert.

Schaltuhr Zeitbereiche haben anders als die Feiertage immer einen Beginn und ein Ende.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

- Wie in der folgenden Methodenbeschreibung eingetragen.

Festlegung zur Priorität:

- Größerer Zahlen bedeuten höhere Gewichtung (Priorität):
- | | |
|------|---|
| 0 | Standardgewichtung von normalen Tagen |
| 1 | Standardgewichtung von Zeitbereichstagen (Ferien) |
| 2 | Standardgewichtung von Feiertagen |
| 3 | Standardgewichtung von Sondertagen |
| 4..9 | Hochpriore Sondertage für Spezialfälle |

Bei gleichen Prioritäten gilt die Reihenfolge Zeitbereich, SondertagJaehrlich und Sondertag Aufzaehlung (höchste Priorität).

Zeitbereich (1:664)

Zeitbereich		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDArt	Zuordnung zum Versorgungsblock.
	Name: bezeichnungType	Name, z.B. Sommerferien. Der Name muss nicht eindeutig sein. (ist nur für Rücklesen der Versorgung in OTEC Format nötig). Falls in Instations nicht angegeben, hier Leerstring eintragen.
	Wochenplan.	Referenz auf verwendeten Wochenplan
	Prioritaet	Wertebereich 1-9, Defaultwert Ferien (Zeitbereiche) = 1
	Start	
	Tag: ui1	Starttag des Zeitbereichs
	Monat: ui1	Startmonat des Zeitbereichs
	Jahr	Startjahr des Zeitbereichs. NULLVALUE (0xffff) bedeutet jährlicher Bereich. In diesem Fall muss auch das Endjahr auf NULLVALUE gesetzt werden.
	Ende	
	Tag: ui1	Endtag des Zeitbereichs
	Monat: ui1	Endmonat des Zeitbereichs

Zeitbereich		
METHOD	Name	Beschreibung
	Jahr	Endjahr des Zeitbereichs. NULLVALUE (0xffff) bedeutet jährlicher Bereich. NULLVALUE (0xffff) bedeutet jährlicher Bereich. In diesem Fall muss auch das Endjahr auf NULLVALUE gesetzt werden.

Hinweis: Alle Zeitbereiche, auch jährlich wiederkehrende, können auch über den Jahreswechsel hinausgehen, z.B. vom 23.12. bis 06.01.

3.3.4 Block 3: VA-Steuerverfahren

3.3.4.1 Objekt BinaerVASteuerverfahren

In diesem Objekt werden die nicht in OCIT-O standardisierten Daten von VA-Steuerverfahren übertragen.

Das Steuergerät erkennt anhand der Kennungen „Member“, „Kennung“ und „DatenBinaer.Typ“ (definiert vom Hersteller des Steuerverfahrens) den Typ der Daten und verarbeitet diese weiter. Die Kennungen müssen vom Lieferanten des VA-Verfahrens mitgeteilt werden. Dem Steuergerät nicht bekannte Daten werden abgelehnt und führen zu einem Versorgungsfehler.

Festlegung zur Reihenfolge in der die Daten sortiert und gelesen werden:

- DatenBinaer Element sortiert aufsteigend nach dem ASCII-Wert der Zeichen von Begin bis Ende des DatenBinaer.Typ Strings.

BinaerVASteuerverfahren (1:672)

BinaerVASteuerverfahren		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDart	Zuordnung zum Versorgungsblock.
	Member	OCIT Memberkennung Kennung des Herstellers des Steuerverfahrens
	VerfahrensVersion: kurzStringType	OCIT-I VerfahrensVersion Text: Hersteller des Steuerverfahrens
	Anzahl	Anzahl folgender DatenBinaer Elemente [1 ... 254].
	DatenBinaer.Typ:bezeichnungType	Bezeichnung des herstellerspezifischen Typs. Text: Hersteller des Steuerverfahrens

BinaerVASteuerverfahren		
METHOD	Name	Beschreibung
	DatenBinaer.Daten:WERT_BLOB	Daten Daten in binärer Darstellung. Sollte base64 Encoding erforderlich sein, ist das vom VD-Server zu erledigen!

3.3.5 Block 4: VA-Parameter

3.3.5.1 Objekt BinaerVAParameter

In diesem Objekt werden die nicht in OCIT-O standardisierten Daten von VA-Parametern übertragen.

Das Steuergerät erkennt anhand der Kennungen „Member“, „Kennung“ und „DatenBinaer.Typ“ (definiert vom Hersteller des Steuerverfahrens) den Typ der Daten und verarbeitet diese weiter. Die Kennungen müssen vom Lieferanten des VA-Verfahrens mitgeteilt werden. Dem Steuergerät nicht bekannte Daten werden abgelehnt und führen zu einem Versorgungsfehler.

Festlegung zur Reihenfolge in der die Versorgungsdaten sortiert und gelesen werden:

- DatenBinaer Element sortiert aufsteigend nach dem ASCII-Wert der Zeichen von Begin bis Ende des DatenBinaer.Typ Strings.

BinaerVAParameter (1:676)

BinaerVAParameter		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	VDart	Zuordnung zum Versorgungsblock.
	Member	OCIT Memberkennung Kennung des Herstellers des Steuerverfahrens
	VerfahrensVersion: kurzStringType	OCIT-I VerfahrensVersion Text: Hersteller des Steuerverfahrens
	Anzahl	Anzahl folgender DatenBinaer Elemente [1 ... 254].
	DatenBinaer.Typ:bezeichnungType	Bezeichnung des herstellerepezifischen Typs.
	DatenBinaer.Daten:WERT_BLOB	Daten. Daten in binärer Darstellung. Sollte base64 Encoding erforderlich sein, ist das vom VD-Server zu erledigen!

3.4 Zentrale Schaltwünsche

Ein zentraler Bediener kann folgende Schaltvorgänge automatisch oder manuell veranlassen:

- Gesamtknoten ein- ausschalten
- Lokale Gesamtknoten Ein-/Ausschaltung freigeben. Der Auszustand kann sein: Aus-Dunkel oder Aus-Blinken (RiLSA und Sonderblinken).
- Zentrales Signalprogramm wählen (max. 255); lokale Signalprogrammwahl freigeben
- Teilknoten wie Gesamtknoten schaltenoder ausschalten (in Auszustand).
- Lokale Schaltung der Teilknoten freigeben. Der Ist-Zustand eines Teilknotens kann ein oder aus (in Auszustand) sein. Der Hauptknoten lässt sich nicht über diesen Mechanismus schalten. Der Auszustand kann sein: Aus-Dunkel oder Aus-Blinken (RiLSA und Sonderblinken).
- Verkehrsabhängigkeit ein-, ausschalten, lokale Schaltung der Verkehrsabhängigkeit freigeben.
- Sondereingriff x ein-, ausschalten, lokalen Sondereingriff freigeben.

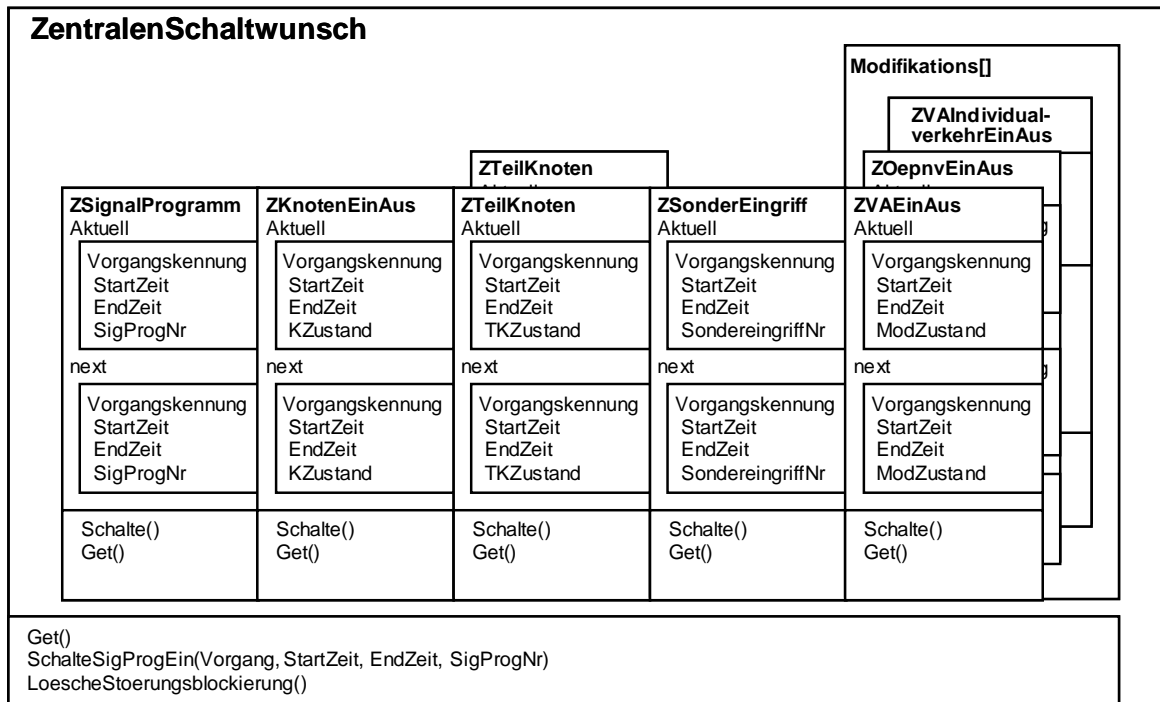
Für jedes Schaltobjekt ist neben dem eigentlichen Schaltwert und der Vorgangskennung ein Gültigkeitszeitraum vorgesehen, der in Form einer Start- und einer Endzeit vorgegeben wird. (Auflösung eine Sekunde).

Die **Startzeit** ermöglicht den Ausgleich unterschiedlicher Übertragungszeiten für das synchrone Schalten mehrerer Lichtsignalsteuergeräte. Schaltwünsche gelten erst mit Erreichen der Startzeit; bis dahin bleibt der neue Wunsch in Warteposition und der alte aktuell. Ein in der Zukunft liegender Schaltwunsch überschreibt immer den in Warteposition. Pro Schalteinheit gibt es zwei Schaltvorgänge zur Speicherung der Zentralschaltwünsche, einen für den aktuellen und einen für den nächsten Schaltvorgang. Ein in der Zukunft liegender Schaltwunsch überschreibt immer den nächsten Schaltvorgang.

Auf Grund der möglichen Zeitdifferenzen (zulässig ist +- 500 ms) zwischen Gerät und Zentrale sollte bei Schaltwünschen, die sofort gültig werden sollen, die Startzeit ausreichend in der Vergangenheit liegen.

Die **Endzeit** ermöglicht grundsätzlich das Zurückschalten auf lokalen Betrieb zur definierten Uhrzeit ohne weitere Verbindung zur Zentrale. Jeder vom Gerät akzeptierte Zentralschaltwunsch bleibt im Gerät, unabhängig von etwaigen Störungen des Übertragungswegs, bis zu seiner Endzeit gültig.

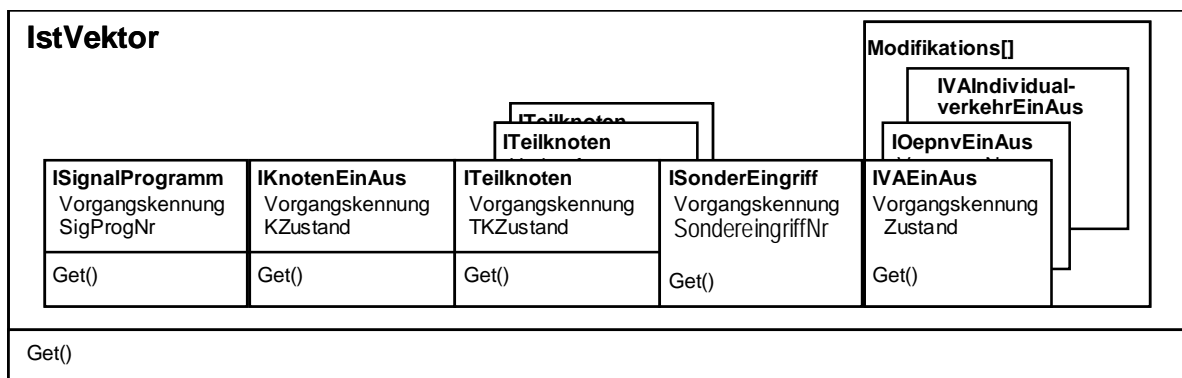
Folgende Bedienvorgänge werden im Gerät als **Schaltwünsche** der Zentrale gespeichert:



Obiges Modell geht davon aus, dass jedes Steuerelement des Zentralenschaltwunsches unabhängig von den anderen Steuerelementen einstellbar ist. Schaltet z.B. die Zentrale ein neues Signalprogramm, so bleiben die anderen Schaltwünsche, also auch ZKnotenEinAus und die Modifikationen unverändert erhalten. Die Schaltwünsche haben die der Zentrale zugeordnete Priorität gegenüber lokalen Schaltwünschen. Gibt die Zentrale z.B. die lokale Signalprogrammwahl frei, wählt das Lichtsignalsteuergerät das Signalprogramm aufgrund anderer lokaler Kriterien (gemäß Schaltuhr oder fest eingestellter Ortsplan). OCIT-konforme Konstellationen der Schaltwünsche siehe Pkt. 3.4.1.

Die Vorgangskennung dient der Zentrale zur Zuordnung der Schaltung zu Bedienern und Gruppenschaltungen. Bei Schaltungen in den Zentralenschaltwunsch überträgt das Gerät mit dem Erreichen des gewünschten Zustandes, diesen und dessen Vorgangskennung in das zugehörige IstVektorelement. Zu jedem Steuerelement gibt es ein Statusselement im IstVektor. Der IstVektor und seine Statusselemente sind einzeln oder gesamt nur lesbar (nicht beschreibbar). Er dient der Beobachtung des am Gerät eingestellten Zustands.

Schema des IstVektors:



3.4.1 OCIT-O konforme Konstellationen der zentralen Schaltwünsche

In der OCIT-Definition sind sowohl der Signalplan, der Gesamtknotenzustand und auch der Zustand der einzelnen Teilknoten mit eigenen Methoden und damit verbunden auch unterschiedlichen Gültigkeitsdauern sehr flexibel einstellbar. Um ein eindeutiges Geräteverhalten zu erreichen, werden hier OCIT-konforme Konstellationen angegeben.

Folgende Zustände sind für die oben genannten Objekte definiert:

Objekt	Zulässige Werte	Bedeutung
Signalprogramm	0 1-255	<p>Von der Zentrale kein Signalprogramm gewählt, daraus folgt lokale Signalprogrammwahl.</p> <p>Auswahl des entsprechenden Signalprogramms. Falls dieses im Gerät nicht definiert ist, erfolgt eine Fehlerquittung und es wird kein neuer Schaltwunsch eingetragen, d.h. der alte Zustand bleibt bestehen.</p>
KZustand	0 - 5	<p>0 Keiner = Kein Zentralenschaltwunsch bezüglich Ein oder Auszustand d. h. Freigabe der lokalen KZustandswahl bzw. unbekannter Ein-/Auszustand des Knotens.</p> <p>1 Gesamtknoten Ein = Knoten ist in das durch Objekt ZSignalProgramm angegebene Signalprogramm einzuschalten bzw. für alle eingeschalteten Teilknoten gilt, dass das durch ISignalProgramm angegebene Signalprogramm bearbeitet wird.</p> <p>2 Gesamtknoten AusDefault = Knoten ist nach Aus Default zu schalten bzw. für alle eingeschalteten Teilknoten gilt: durch Versorgung ausgewählte Signalgruppen, normalerweise die Fahrzeugsignalgruppen der Nebenrichtung, blinken, die restlichen Signalgruppen sind dunkel.</p> <p>3 Gesamtknoten AusBlinkenNebenrichtung = Knoten ist nach Aus-Blinken-Nebenrichtung zu schalten bzw. für alle eingeschalteten Teilknoten gilt: Die Fahrzeugsignalgruppen der Nebenrichtung, blinken, die restlichen Signalgruppen sind dunkel.</p> <p>4 Gesamtknoten AusDunkel = Knoten ist nach Aus-Dunkel zu schalten bzw. alle Signalgruppen der eingeschalteten Teilknoten sind dunkel.</p> <p>5 Gesamtknoten AusBlinkenAlle = Knoten ist nach Aus Blinken alle zu schalten bzw. alle Signalgruppen der eingeschalteten Teilknoten blinken.</p> <p>6 – 255 nicht zulässig</p>

Objekt	Zulässige Werte	Bedeutung
TKZustand	0 - 5	<p>0 keiner = kein Zustand eingestellt oder unbestimmter Zustand</p> <p>1 Ein = bedeutet</p> <ul style="list-style-type: none"> • bei Verwendung als Schaltbefehl: Teilknoten ist in den KZustand des Gesamtknotens zu schalten • bei Verwendung als Ist-Zustand: Teilknoten ist eingeschaltet. <p>2 AusDefault = Durch Versorgung ausgewählte Signalgruppen, normalerweise die Fahrzeugsignalgruppen der Nebenrichtung, blinken, die restlichen Signalgruppen sind dunkel.</p> <p>3 AusBlinkenNebenrichtung = Alle Signalgruppen der Nebenrichtung des Teilknotens blinken.</p> <p>4 AusDunkel = Alle Signalgruppen des Teilknotens sind dunkel.</p> <p>5 AusBlinkenAlle = Alle Signalgruppen des Teilknotens blinken.</p> <p>6 – 255 = nicht zulässig</p>

Wenn die Gültigkeitsdauer eines einzelnen Schaltwunsches ausläuft und kein weiterer gültiger entsprechender Schaltwunsch vorliegt, dann fällt der jeweilige Schaltwunsch in den Zustand 0 mit einer unbegrenzten Gültigkeitsdauer zurück.

KZustand = 0 (oder TKZustand = 0) bedeutet, dass der Gesamtknoten (oder Teilknoten) so sein soll wie aktuell in der lokalen Zeitsteuerung hinterlegt, das Gerät aber trotzdem im Zentralbetrieb mit dem gewünschten Signalplan läuft.

Auf Grund der Schaltsystematik von OCIT-O können Schaltzustände erzeugt werden, die mehrdeutig interpretiert werden können. Ab der Version OCIT-O V2.0 werden diese eindeutig festgelegt.

Signalprogramm	KZustand	TKZustand	Reaktion	IstVektor *)
0	0	0	Betrieb gemäß lokaler Steuerung	Lokal 0 bis 4
		1	Betrieb gemäß lokaler Steuerung	Lokal 0 bis 4
		2-5	Betrieb gemäß lokaler Steuerung, aber TK-Zustand entsprechend zentralen TK-Zustand.	Zentrale
	1	0	Signalplanauswahl gemäß lokaler Steuerung. Gesamtknoten zentral. Achtung Sonderfall: Es kann sein dass alle Teilknoten durch die lokale JAUT ausgeschaltet werden obwohl der Gesamtknoten eingeschaltet ist.	Zentrale
		1	Signalplanauswahl gemäß lokaler Steuerung. Knotenzustände zentral.	Zentrale
		2-5	Betrieb gemäß lokaler Steuerung, aber TK-Zustand entsprechend zentralen TK-Zustand.	Zentrale
	2-5	0	Auszustand gemäß zentralen Knoten-Zustand	Zentrale
		1	Auszustand gemäß zentralen Knoten-Zustand	Zentrale
		2-5	Auszustand der einzelnen TK's wie in zentralen TKZustand hinterlegt	Zentrale
	1-255	0	0	Zentrale Signalprogrammwahl, aber lokaler Knotenzustand.
1			Zentrale Signalprogrammwahl, aber lokaler Knotenzustand.	Zentrale
2-5			Auszustand der einzelnen TK's wie in zentralen TKZustand hinterlegt	Zentrale

Signalprogramm	KZustand	TKZustand	Reaktion	IstVektor *)
	1	0	Zentralenbetrieb Signalprogramm wie hinterlegt	Zentrale
		1	Zentralenbetrieb Signalprogramm wie hinterlegt	Zentrale
		2-5	Zentralenbetrieb Teilknoten im hinterlegten Aus-Zustand	Zentrale
	2-5	0	Zentralenbetrieb Aus - Teilknoten im hinterlegten Aus-Zustand (Kzustand)	Zentrale
		1	Zentralenbetrieb Aus - Teilknoten im hinterlegten Aus-Zustand (Kzustand)	Zentrale
		2-5	Zentralenbetrieb Aus - Teilknoten im hinterlegten Aus-Zustand (TKzustand)	Zentrale

*) Der Zustand der Modifikationen hat keinen Einfluss auf die zurückgemeldete Betriebsart!

Verhalten der Steuergeräte bei Anforderungen von nicht versorgten Signalprogrammen durch eine Zentrale:

- Wird von der Zentrale ein nicht versorgtes Signalprogramm angefordert, so wird der Schaltwunsch mit einer Fehlermeldung (Returncode) zurückgewiesen und nicht vom Steuergerät übernommen. Dies führt dazu, dass nach Ablauf des letzten gültigen Schaltwunsches das Gerät auf lokale Betriebsart zurückfällt.
- Sollte aus dem Gerätezustand Aus der Versuch gemacht werden in ein nicht versorgtes Programm einzuschalten, so muss das Gerät Aus bleiben. Fehlermeldung des Btpl-Protokolls: Param_invalid.

3.4.2 Struktur ZEITINTERVALL

ZEITINTERVALL besteht aus StartZeit und EndZeit. Es ist gültig falls:

$$0 \leq \text{StartZeit} < \text{EndZeit} \text{ und die Endzeit nicht in der Vergangenheit liegt.}$$

Ein Zeitintervall ist aktiv falls es die aktuelle Zeit enthält:

$$0 \leq \text{StartZeit} \leq \text{aktuelle Zeit} \leq \text{EndZeit}$$

3.4.3 Typen und Pfade

Alle Knotenbezogenen Strukturen erhalten als ersten Pfadparameter die relative Knotennummer. Damit ist es möglich mehrere Knoten innerhalb eines Lichtsignalsteuergerätes zu adressieren.

Mit dieser relativen Knotennummer (RelKnotenNr) sehen die Pfade wie folgt aus:ZentralenSchaltwunsch

OType	Pfad (Betreiber()/Vsr()/Lichtsignalsteuergerät() immer im BTPPL Header)
220	ZentralenSchaltwunsch/RelKnotenNr()
222	ZentralenSchaltwunsch/ZSignalProgramm/RelKnotenNr()
224	ZentralenSchaltwunsch/ZKnotenEinAus/RelKnotenNr()
226	ZentralenSchaltwunsch/ZTeilknoten/RelKnotenNr()/TeilknotenNummer()
228	ZentralenSchaltwunsch/ZSonderEingriff/RelKnotenNr()
230	ZentralenSchaltwunsch/ZVAEinAus/RelKnotenNr()
232	ZentralenSchaltwunsch/ZOepnvEinAus/RelKnotenNr()
234	ZentralenSchaltwunsch/ZProjEinAus/RelKnotenNr()/ProjModNr()
238	ZentralenSchaltwunsch/ZVAIndividualverkehrEinAus/RelKnotenNr()

IstVektor

OType	Pfad (Betreiber()/Vsr()/Lichtsignalsteuergerät() immer im BTPPL Header)
221	IstVektor/RelKnotenNr()
223	IstVektor/ISignalProgramm/RelKnotenNr()
225	IstVektor/IKnotenEinAus/RelKnotenNr()
227	IstVektor/ITeilknoten/RelKnotenNr()/TeilknotenNummer()
229	IstVektor/ISonderEingriff/RelKnotenNr()
231	IstVektor/IVAEinAus/RelKnotenNr()
233	IstVektor/IOepnvEinAus/RelKnotenNr()
235	IstVektor/IProjEinAus/RelKnotenNr()/ProjModNr()
239	IstVektor/IVAIndividualverkehrEinAus /RelKnotenNr()

3.4.4 Objekt ZSignalProgramm

ZSignalprogramm speichert den Signalprogramm Schaltwunsch der Zentrale.

ZSignalprogramm (1:222)

ZSignalprogramm		
METHOD	Name	Beschreibung
16	Schalte	<p>Nächsten Signalprogrammschaltwunsch der Zentrale entgegennehmen.</p> <p>Falls das durch Start- und EndZeit angegebene Intervall ungültig oder in der Vergangenheit liegt, kehrt diese Methode mit RetCode = INTERVALL_INVALID zurück.</p> <p>Die Serverfunktion in F prüft ob zur angegebenen SigProgNr ein Signalprogramm versorgt ist oder SigProgNr gleich 0 ist (Freigabe der lokalen Signalprogrammwahl). Falls nein kehrt sie mit RetCode=PARAM_INVALID zurück.</p> <p>Falls das übergebene Intervall die aktuelle Zeit enthält (StartZeit <= aktuelle Zeit < EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur Aktuell, stößt den Signalprogrammwechsel an und kehrt mit RETCODE=OK zurück (wartet nicht bis SigProgNr tatsächlich läuft). Ist der Knoten zur Zeit über 'ZKnotenEinAus' ausgeschaltet, so schaltet er nicht ein bzw. um. Dadurch kann die Zentrale ein Signalprogramm vorgeben und die Ein Ausschaltungen von der lokalen ZAUT vornehmen lassen.</p> <p>Falls das übergebene Intervall in der Zukunft liegt (aktuelle Zeit < StartZeit < EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur next. Ein eventuell in next stehender Schaltauftrag wird storniert und überschrieben.</p> <p>Wird die StartZeit eines in next abgelegten Signalprogrammschaltwunsches erreicht, überträgt F diesen in die Unterstruktur Aktuell, ermittelt und schaltet das dann wirksame Signalprogramm.</p> <p>Wird die EndZeit eines in der Unterstruktur Aktuell abgelegten Schaltwunsches erreicht, beendet F diesen Schaltwunsch. D.h. F ermittelt und schaltet das dann wirksame Signalprogramm.</p> <p>Enthält ZSignalProgramm kein zur Zeit aktives Intervall, liegt kein Zentralen Signalprogramm-Schaltwunsch vor. Die Behandlung entspricht dann einem Schaltwunsch mit SigProgNr=0, d.h. es liegt kein aktueller Signalprogrammschaltwunsch der Zentrale vor.</p>
	Eingabeparameter	
	Vorgang : SYSJOBID	Bedienvorgangskennung des Aufrufers für diese Signalprogrammschaltung.
	StartZeit: utc	Ab „StartZeit“ gilt dieser Signalprogrammschaltwunsch
	EndZeit : utc	Bis „EndZeit“ gilt dieser Signalprogrammschaltwunsch
	SigProgNr : ui1	0 Freigabe der lokalen Signalprogrammwahl 1-255 Signalprogramme.
	Ausgabeparameter	

ZSignalprogramm		
METHOD	Name	Beschreibung
	RetCode	OK: Auftrag angenommen INTERVALL_INVALID: ungültiges oder verstrichenes Zeitintervall angegeben, Auftrag abgelehnt. PARAM_INVALID: ungültige Signalprogrammnummer, Auftrag abgelehnt.
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Aktuell.Vorgang	Vorgangskennung des Vorgangs welcher diese Signalprogrammnummer eingestellt hat.
	Aktuell.StartZeit	Seit diesem Zeitpunkt ist/war dieser Schaltwunsch aktiv.
	Aktuell.EndZeit	Bis zu diesem Zeitpunkt ist/war dieser Schaltwunsch aktiv.
	Aktuell.SigProgNr	Von der Zentrale für obiges Zeitintervall gewünschte Signalprogrammnummer.
	next.Vorgang	Nächster Zentralenschaltwunsch.
	next.StartZeit	
	next.EndZeit	
next.SigProgNr		

3.4.5 Objekt ZKnotenEinAus

Dieses Objekt speichert den Schaltwunsch der Zentrale für den Gesamtknotenstatus:

KZustand	
Name	Bedeutung im Schaltwunsch bzw. IstStand
Keiner =0	Freigabe der lokalen KZustandswahl, kein Zentralenschaltwunsch bezüglich Ein oder Auszustand.
Ein =1	Knoten ist in das durch das angegebene Signalprogramm einzuschalten bzw. für alle eingeschalteten Teilknoten gilt: das durch ISignalProgramm angegebene Signalprogramm wird bearbeitet.
AusDefault =2	Knoten ist nach AusDefault zu schalten. Für alle eingeschalteten Teilknoten gilt: durch Versorgung ausgewählte Signalgruppen, normalerweise die Fahrzeugsignalgruppen der Nebenrichtung, blinken, die restlichen Signalgruppen sind dunkel.
AusBlinken-Nebenrichtung =3	Knoten ist nach Aus BlinkenNebenrichtung zu schalten bzw. für alle eingeschalteten Teilknoten gilt: Die Fahrzeugsignalgruppen der Nebenrichtung, blinken, die restlichen Signalgruppen sind dunkel.
AusDunkel =4	Knoten ist nach AusDunkel zu schalten bzw. alle Signalgruppen der eingeschalteten Teilknoten sind dunkel.

KZustand	
Name	Bedeutung im Schaltwunsch bzw. IstStand
AusBlinkenAlle =5	Knoten ist nach AusBlinken alle zu schalten bzw. alle Signalgruppen der eingeschalteten Teilknoten blinken.
6..255	Reserviert

ZKnotenEinAus steuert die Ein- Ausschaltung des Gesamtknotens. Wird z.B. mittels ZKnotenEinAus der gesamte Knoten eingeschaltet, schalten alle Teilknoten deren TKZustand gleich Ein =1 ist ein.

Prinzipiell könnte die Gesamtknoten Ein/Ausschaltung auch über die ZTeilknoten erfolgen. Da aber die Kreuzungsgeräte verschiedene Abläufe für zu und Abschalten von Teilknoten oder Gesamt ein Ausschalten ausführen, erscheint die direkte Vorgabe von Gesamtknoten Ein/Aus transparenter.

ZKnotenEinAus (1:224)

ZKnotenEinAus		
METHOD	Name	Beschreibung
16	Schalte	<p>Nächsten Ein- Ausschaltwunsch der Zentrale entgegennehmen.</p> <p>Falls das durch Start- und EndZeit angegebene Intervall ungültig oder in der Vergangenheit liegt, kehrt diese Methode mit RetCode = INTERVALL_INVALID zurück.</p> <p>Die Serverfunktion in F prüft der angegebene KZustand gültig ist. Falls nein kehrt sie mit RetCode=PARAM_INVALID zurück.</p> <p>Falls das übergebene Intervall die aktuelle Zeit enthält (StartZeit <= aktuelle Zeit <EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur Aktuell, stößt die Ein- Ausschaltung an und kehrt mit RETCODE=OK zurück (wartet nicht bis tatsächlich geschaltet ist).</p> <p>Falls das übergebene Intervall in der Zukunft liegt (aktuelle Zeit < StartZeit < EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur next. Ein eventuell in next stehender Schaltauftrag wird storniert und überschrieben.</p> <p>Wird die StartZeit eines in next abgelegten Schaltwunsches erreicht, überträgt F diesen in die Unterstruktur Aktuell, ermittelt und schaltet den dann wirksamen KZustand.</p> <p>Wird die EndZeit eines in der Unterstruktur Aktuell abgelegten Schaltwunsches erreicht, beendet F diesen Schaltwunsch. D.h. F ermittelt und schaltet den dann wirksamen KZustand.</p> <p>Enthält ZKnotenEinAus kein zur Zeit aktives Intervall, liegt kein Zentralen EinAus-Schaltwunsch vor. Die Behandlung entspricht dann einem Schaltwunsch mit KZustand=0, d.h. es liegt kein aktueller Ein- Ausschaltwunsch der Zentrale vor.</p> <p>Jedes Gerät muss in den Knotenzustand AusDefault schalten können. Kann ein Gerät nicht in AusBlinkenNebenrichtung, AusDunkel oder AusBlinkenAlle schalten, so schaltet es statt dessen in AusDefault. In diesem Fall meldet das Gerät im IKnotenEinAus AusDefault, in ZKnotenEinAus wird jedoch der übergebene KZustand eingetragen.</p>
Eingabeparameter		
	Vorgang : SYSJOBID	Bedienvorgangskennung des Aufrufers für diese Signalprogrammumschaltung.
	StartZeit: utc	Ab „StartZeit“ gilt dieser Ein- Ausschaltwunsch
	EndZeit : utc	Bis „EndZeit“ gilt dieser Ein- Ausschaltwunsch
	KZustand : ui1	Siehe KZustand
Ausgabeparameter		
	RetCode	<p>OK: Auftrag angenommen</p> <p>INTERVALL_INVALID: ungültiges oder verstrichenes Zeitintervall angegeben, Auftrag abgelehnt.</p> <p>PARAM_INVALID: ungültiger KZustand, Auftrag abgelehnt.</p>

ZKnotenEinAus		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Aktuell.Vorgang	
	Aktuell.StartZeit	
	Aktuell.EndZeit	
	Aktuell.KZustand	
	next.Vorgang	
	next.StartZeit	
	next.EndZeit	
	next.KZustand	

3.4.6 Objekt ZTeilKnoten

ZTeilKnoten speichert den Schaltwunsch der Zentrale für einen Teilknoten:

TKZustand	
Name	Bedeutung im Schaltwunsch bzw. IstStand
Keiner =0	Freigabe der lokalen TKZustandswahl, kein Zentralenschaltwunsch bezüglich TKEin-/ Auszustand bzw. unbekannter Zustand.
Ein =1	Verwendung als Schaltbefehl: Teilknoten ist in den KZustand des Gesamtknotens zu schalten Verwendung als Ist-Zustand: Teilknoten ist eingeschaltet.
AusDefault =2	Teilknoten ist Aus default, unabhängig vom KZustand des Gesamtknotens. D. h. durch Versorgung ausgewählte Signalgruppen, normalerweise die Fahrzeugsignalgruppen der Nebenrichtung, blinken, die restlichen Signalgruppen sind dunkel.
AusBlinken-Nebenrichtung =3	Die Fahrzeugsignalgruppen der Nebenrichtung, blinken, die restlichen Signalgruppen sind dunkel, unabhängig vom KZustand des Gesamtknotens.
AusDunkel =4	Alle Signalgruppen des Teilknotens sind dunkel, unabhängig vom KZustand des Gesamtknotens.
AusBlinkenAlle =5	Alle Signalgruppen des Teilknotens blinken, unabhängig vom KZustand des Gesamtknotens.
6..255	Reserviert

Um Methoden einer Instanz ZTeilKnoten aufzurufen sind im Feld Path (siehe OCIT-O Protokoll) die gewünschte relative Knoten- und Teilknotennummer jeweils als ui1 anzugeben.

ZTeilknoden (1:226)

ZTeilknoden				
METHOD	Name	Beschreibung		
16	Schalte	Nächsten Teilknoden Ein- Ausschaltwunsch der Zentrale entgegennehmen.		
		Falls das durch Start- und EndZeit angegebene Intervall ungültig oder in der Vergangenheit liegt, kehrt diese Methode mit RetCode = INTERVALL_INVALID zurück.		
		Die Serverfunktion in F prüft der angegebene TKZustand gültig ist. Falls nein kehrt sie mit RetCode=PARAM_INVALID zurück.		
		Falls das übergebene Intervall die aktuelle Zeit enthält (StartZeit <= aktuelle Zeit <EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur Aktuell, stößt die Teilknoden Ein- Ausschaltung an und kehrt mit RETCODE=OK zurück (wartet nicht bis tatsächlich geschaltet ist).		
		Falls das übergebene Intervall in der Zukunft liegt (aktuelle Zeit < StartZeit < EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur next. Ein eventuell in next stehender Schaltauftrag wird storniert und überschrieben.		
		Wird die StartZeit eines in next abgelegten Schaltwunsches erreicht, überträgt F diesen in die Unterstruktur Aktuell, ermittelt und schaltet den dann wirksamen TKZustand.		
		Wird die EndZeit eines in der Unterstruktur Aktuell abgelegten Schaltwunsches erreicht, beendet F diesen Schaltwunsch. D.h. F ermittelt und schaltet den dann wirksamen TKZustand.		
		Enthält ZTeilKnoden kein zur Zeit aktives Intervall, liegt kein Zentralen Teilknoden-Schaltwunsch vor. Die Behandlung entspricht dann einem Schaltwunsch mit TKZustand=0, d.h. es liegt kein aktueller Teilknoden Ein- Ausschaltwunsch der Zentrale vor.		
		Eingabeparameter		
		Vorgang : SYSJOBID	Bedienvorgangskennung des Aufrufers für diese Signalprogrammumschaltung.	
StartZeit: utc	Ab „StartZeit“ gilt dieser Teilknoden Ein- Ausschaltwunsch			
EndZeit : utc	Bis „EndZeit“ gilt dieser Teilknoden Ein- Ausschaltwunsch			
TKZustand : ui1	Siehe TKZustand			
Ausgabeparameter				
RetCode	OK: Auftrag angenommen INTERVALL_INVALID: ungültiges oder verstrichenes Zeitintervall angegeben, Auftrag abgelehnt. PARAM_INVALID: ungültiger TKZustand, Auftrag abgelehnt. PATH_INVALID: keine oder ungültige Teilknodenummer im Path Feld angegeben, Auftrag abgelehnt.			
0	Get			
	Ausgabeparameter			

ZTeilknoten		
METHOD	Name	Beschreibung
	RetCode	OK: folgende Parameter richtig gelesen
	Aktuell.Vorgang	
	Aktuell.StartZeit	
	Aktuell.EndZeit	
	Aktuell.TKZustand	
	next.Vorgang	
	next.StartZeit	
	next.EndZeit	
	next.TKZustand	

3.4.7 Objekt ZSondereingriff

ZSondereingriff speichert den Schaltwunsch der Zentrale für Sondereingriffe ab. Die darin enthaltene SonderEingriffNr hat folgende Bedeutungen:

SonderEingriffNr	
Wert	Bedeutung
0	Freigabe lokaler Sondereingriffe, kein Sondereingriff.
1..254	Temporär gültiges Signalprogramm, z.B. Feuerwehrplan Route 1..n.
255	Sondereingriff Aus, Blockierung lokaler Sondereingriffe.

Liegt ein zentralen Signalprogrammschaltwunsch (ZSignalProgramm) und ein Sondereingriff Schaltwunsch für die gleiche Zeit an, so schaltet das Gerät den Sondereingriff, jedoch nur falls der Knoten eingeschaltet ist (ZKnotenEinAus).

Das Objekt ZSondereingriff ist nötig, damit das Gerät nach Ablauf des Sondereingriff ohne weitere Zentralenkommunikation wieder in das normale zentralen Signalprogramm schalten kann.

ZSondereingriff (1:228)

ZSondereingriff		
METHOD	Name	Beschreibung
16	Schalte	<p>Nächsten Sonder-Signalprogrammschaltwunsch der Zentrale entgegennehmen.</p> <p>Falls das durch Start- und EndZeit angegebene Intervall ungültig oder in der Vergangenheit liegt, kehrt diese Methode mit RetCode = INTERVALL_INVALID zurück.</p> <p>Die Serverfunktion in F prüft ob zur angegebenen SonderEingriffNr ein Signalprogramm versorgt ist. Falls nein kehrt sie mit RetCode=PARAM_INVALID zurück.</p> <p>Falls das übergebene Intervall die aktuelle Zeit enthält (StartZeit <= aktuelle Zeit <EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur Aktuell, stößt den Signalprogrammwechsel an und kehrt mit RETCODE=OK zurück (wartet nicht bis SonderEingriffNr tatsächlich läuft).</p> <p>Falls das übergebene Intervall in der Zukunft liegt (aktuelle Zeit < StartZeit < EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur next. Ein eventuell in next stehender Schaltauftrag wird storniert und überschrieben.</p> <p>Wird die StartZeit eines in next abgelegten Sonder- Signalprogramm-schaltwunsches erreicht, überträgt F diesen in die Unterstruktur Aktuell, ermittelt und schaltet das dann wirksame Signalprogramm.</p> <p>Wird die EndZeit eines in der Unterstruktur Aktuell abgelegten Schaltwunsches erreicht, beendet F diesen Schaltwunsch. D.h. F ermittelt und schaltet das dann wirksame Signalprogramm.</p> <p>Enthält ZSondereingriff kein zur Zeit aktives Intervall, liegt kein Zentrales Sonder-Signalprogramm-Schaltwunsch vor. Die Behandlung entspricht dann einem Schaltwunsch mit SonderEingriffNr =0, d.h. es liegt kein aktueller Sonder-Signalprogrammschaltwunsch der Zentrale vor.</p>
	Eingabeparameter	
	Vorgang : SYSJOBID	Bedienvorgangskennung des Aufrufers für diese Signalprogramm-schaltung.
	StartZeit: utc	Ab „StartZeit“ gilt dieser Sonder-Signalprogrammschaltwunsch
	EndZeit : utc	Bis „EndZeit“ gilt dieser Sonder-Signalprogrammschaltwunsch
	SonderEingriffNr : ui1	Siehe SonderEingriffNr.
	Ausgabeparameter	
	RetCode	<p>OK: Auftrag angenommen</p> <p>INTERVALL_INVALID: ungültiges oder verstrichenes Zeitintervall angegeben, Auftrag abgelehnt.</p> <p>PARAM_INVALID: ungültige Sonder-Signalprogrammnummer, Auftrag abgelehnt.</p>
0	Get	

ZSondereingriff		
METHOD	Name	Beschreibung
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Aktuell.Vorgang	
	Aktuell.StartZeit	
	Aktuell.EndZeit	
	Aktuell. SonderEin- griffNr	
	next.Vorgang	
	next.StartZeit	
	next.EndZeit	
	next. SonderEin- griffNr	

3.4.8 Signalprogramm Modifikationen

Es gibt einige ein-/ ausschaltbare Signalprogramm Parameter. ModEinAusZustand gibt die Codierung des von der Zentrale gewünschten Zustands einer Modifikation an:

ModEinAusZustand	
Wert	Bedeutung
Keiner =0	Kein Zustand eingestellt, unbestimmter Zustand oder lokale Zustandwahl freigegeben
Aus =1	Modifikation ist ausgeschaltet.
Ein =2	Modifikation ist eingeschaltet

Die Semantik von ModEinAusZustand ist abhängig vom Objekt in welchem es verwendet wird.

Modifikationen werden wirksam in Abhängigkeit von der Priorität der Schaltquelle:

1. Höchste Priorität Manueller Eingriff
2. Zentrale
3. Niedrigste Priorität: Schaltuhr

3.4.8.1 Objekt ZModEinAus

Das Objekt ZModEinAus dient als Basisklasse für alle Modifikationen.

ZModEinAus (1:206)

ZModEinAus		
METHOD	Name	Beschreibung
16	Schalte	<p>Nächsten Modifikations Ein- Ausschaltwunsch der Zentrale entgegennehmen.</p> <p>Falls das durch Start- und EndZeit angegebene Intervall ungültig oder in der Vergangenheit liegt, kehrt diese Methode mit RetCode = INTERVALL_INVALID zurück.</p> <p>Die Serverfunktion in F prüft der angegebene VAZustand gültig ist. Falls nein kehrt sie mit RetCode=PARAM_INVALID zurück.</p> <p>Falls das übergebene Intervall die aktuelle Zeit enthält (StartZeit <= aktuelle Zeit <EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur Aktuell, stößt die Ein- Ausschaltung an und kehrt mit RETCODE=OK zurück (wartet nicht bis tatsächlich geschaltet ist).</p> <p>Falls das übergebene Intervall in der Zukunft liegt (aktuelle Zeit < StartZeit < EndZeit) überträgt diese Methode die angegebenen Parameter in die Unterstruktur next. Ein eventuell in next stehender Schaltauftrag wird storniert und überschrieben.</p> <p>Wird die StartZeit eines in next a gelegten Schaltwunsches erreicht, überträgt F diesen in die Unterstruktur Aktuell, ermittelt und schaltet den dann wirksamen Zustand.</p> <p>Wird die EndZeit eines in der Unterstruktur Aktuell abgelegten Schaltwunsches erreicht, beendet F diesen Schaltwunsch. D.h. F ermittelt und schaltet den dann wirksamen Zustand.</p> <p>Enthält ModEinAus kein zur Zeit aktives Intervall, liegt kein Zentralen EinAus-Schaltwunsch vor. Die Behandlung entspricht dann einem Schaltwunsch mit Zustand=0, d.h. es liegt kein aktueller Ein- Ausschaltwunsch der Zentrale vor.</p>
Eingabeparameter		
	Vorgang : SYSJOBID	Bedienvorgangskennung des Aufrufers für diese Signalprogrammumschaltung.
	StartZeit: utc	Ab „StartZeit“ gilt dieser Ein- Ausschaltwunsch
	EndZeit : utc	Bis „EndZeit“ gilt dieser Ein- Ausschaltwunsch
	Zustand : ModeEinAusZustand	Einzustellender Zustand siehe ModeEinAusZustand.
Ausgabeparameter		

ZModEinAus		
METHOD	Name	Beschreibung
	RetCode	OK: Auftrag angenommen INTERVALL_INVALID: ungültiges oder verstrichenes Zeitintervall angegeben, Auftrag abgelehnt. PARAM_INVALID: ungültiger VAZustand, Auftrag abgelehnt. NOT_CONFIGURED: Angegebene Modifikation ist nicht versorgt, Auftrag abgelehnt.
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Aktuell.Vorgang	Bedienvorgangskennung des Aufrufers für diese Modifikationsschaltung
	Aktuell.StartZeit	Ab diesem Zeitpunkt war dieser Schaltwunsch gültig
	Aktuell.EndZeit	Bis zu diesem Zeitpunkt ist oder war dieser Schaltwunsch gültig
	Aktuell.Zustand	Eingestellter Zustand s. ModEinAusZustand
	Next.Vorgang	Zeitlich nächster Zentralenschaltwunsch
	Next.StartZeit	
	Next.EndZeit	
	Next.Zustand	

3.4.8.2 Objekt IModEinAus

Das Objekt IModEinAus zeigt den aktuell eingestellter Modifikations-EinAusZustand und zugehörige Vorgangskennung. Bedeutung je nach abgeleiteter Klasse VA, ÖPNV, Orientierungstakt ...

IModEinAus (1:207)

IModEinAus		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden Ein- Ausschaltung der Modifikation geführt hat.
	Zustand	Ein- Auszustand dieser Modifikation s. ModeEinAuszustand.

Für alle Modifikationen wird je eine Spezialisierung von ZModEinAus und IModEinAus definiert. Das hat den Vorteil, dass (wenn später nötig) auch Modifikationen mit Parametern machbar sind. Für Alle Modifikationen gilt: Solange im Gerät eine Modifikation nicht implementiert ist,

- liefern die Methoden ZModEinAus.Schalte(..), ZModEinAus.Get(), IModEinAus.Get() einen Fehler zurück (ERR_TYPE oder NOT_CONFIGURED),
- wird IModEinAus weder im IstVektor.Modifikationen[] noch ZModEinAus im ZentralenSchaltwunsch.Get(..Modifikationen[]) übertragen.

3.4.8.3 Objekt ZVAEinAus

(1:230)

ZVAEinAus ist eine Spezialisierung von ZModeEinAus und speichert den Schaltwunsch der Zentrale für den übergeordneten Zustand der lokalen Verkehrsabhängigkeit.

0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Aktuell.Vorgang	Bedienvorgangskennung des Aufrufers für diese Modifikationsschaltung
	Aktuell.StartZeit	Ab diesem Zeitpunkt war dieser Schaltwunsch gültig
	Aktuell.EndZeit	Bis zu diesem Zeitpunkt ist oder war dieser Schaltwunsch gültig
	Aktuell.Zustand	Eingestellter Zustand siehe Tabelle VAZustand
	Next.Vorgang	Zeitlich nächster Zentralenschaltwunsch
	Next.StartZeit	
	Next.EndZeit	
	Next.Zustand	

Der aktuelle Zustand hat im Objekt ZVAEinAus folgende Bedeutung:

VAZustand	
Wert	Bedeutung
Keiner =0	Freigabe der lokalen VA-Zustandswahl.
Aus =1	Aus: Die lokale verkehrsabhängige Logik arbeitet nicht, d.h. Festzeitbetrieb
Ein =2	Ein: Die lokale verkehrsabhängige Logik arbeitet.

Bei aktiviertem Zustand der Verkehrsabhängigkeit kann durch die Objekte ZVAIndividualverkehrEinAus und ZOepnvEinAus das Verhalten der verkehrsabhängigen Logik detailliert werden.

3.4.8.4 Objekt ZOepnvEinAus

(1:232)

Falls ein Gerät bzw. Knoten keine ein- ausschaltbare ÖPNV Bevorzugung unterstützt, liefert die Methode Schalte einen Fehler zurück (ERR_TYPE, NOT_CONFIGURED).
siehe oben (allgem. Modifikation)

ZOepnvEinAus ist eine Spezialisierung von **ZModEinAus** und speichert den von der Zentrale eingestellten übergeordneten Zustand der lokalen ÖPNV Bevorzugung.

0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Aktuell.Vorgang	Bedienvorgangskennung des Aufrufers für diese Modifikationsschaltung
	Aktuell.StartZeit	Ab diesem Zeitpunkt war dieser Schaltwunsch gültig
	Aktuell.EndZeit	Bis zu diesem Zeitpunkt ist oder war dieser Schaltwunsch gültig
	Aktuell.Zustand	Eingestellter Zustand siehe Tabelle OepnvEinAus
	Next.Vorgang	Zeitlich nächster Zentralenschaltwunsch
	Next.StartZeit	
	Next.EndZeit	
	Next.Zustand	

Der aktuelle Zustand hat im Objekt OepnvEinAus folgende Bedeutung:

OepnvEinAus	
Wert	Bedeutung
Keiner = 0	Freigabe der lokalen ÖPNV Bevorzugung.
Aus = 1	Aus: Die lokale ÖPNV Bevorzugung arbeitet nicht
Ein = 2	Ein: Die lokale ÖPNV Bevorzugung arbeitet.

Bei abgeschalteter ÖPNV-Beschleunigung führen Anforderungen des ÖPNV nicht zur Beeinflussung der Signalisierung, d.h. der ÖPNV wird nicht beschleunigt.

3.4.8.4.1 Objekt ZVAIndividualverkehrEinAus

(1:238)

Falls ein Gerät bzw. Knoten keine separate Steuerung der VA unterstützt, liefert die Methode Schalte einen Fehler zurück (ERR_TYPE, NOT_CONFIGURED).
siehe oben (allgem. Modifikation).

ZVAIndividualverkehrEinAus ist eine Spezialisierung von **ZmodEinAus** und speichert den von der Zentrale eingestellten Zustand der Beeinflussung der lokalen verkehrsabhängigen Logik des Knotens durch den Individualverkehr.

0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Aktuell.Vorgang	Bedienvorgangskennung des Aufrufers für diese Modifikationsschaltung
	Aktuell.StartZeit	Ab diesem Zeitpunkt war dieser Schaltwunsch gültig
	Aktuell.EndZeit	Bis zu diesem Zeitpunkt ist oder war dieser Schaltwunsch gültig
	Aktuell.Zustand	Eingestellter Zustand siehe Tabelle ZVAIndividualverkehrEinAus
	Next.Vorgang	Zeitlich nächster Zentralenschaltwunsch
	Next.StartZeit	
	Next.EndZeit	
	Next.Zustand	

Der aktuelle Zustand hat im Objekt ZVAIndividualverkehrEinAus folgende Bedeutung:

ZVAIndividualverkehrEinAus	
Wert	Bedeutung
Keiner = 0	Freigabe der VA-Beeinflussung durch Individualverkehr / unbekannter Zustand.
Aus = 1	Aus: Individualverkehr beeinflusst VA nicht (VA-Reduzierung)
Ein = 2	Ein: Individualverkehr beeinflusst VA

Befindet sich die Modifikation im Zustand „AUS“, so beeinflussen Ereignisse des Individualverkehrs, wie z.B. Detektoren, nicht die verkehrstechnische Logik. Dieser Zustand wird als Reduzierung der VA bezeichnet.

3.4.8.5 Kombination von Modifikationen

Aus der Kombination der Modifikationen lässt sich die Verarbeitung von verkehrstechnischen Anforderungen beeinflussen. Die mögliche Bedeutung der Kombinationen bei Verwendung aller drei Kombinationen zeigt folgende Tabelle. Es wird die mögliche Beeinflussung der verkehrstechnischen Logik durch den Individualverkehr und ÖPNV abgebildet.

VA EinAus	VAIndividualverkehr EinAus	ÖPNV EinAus	Geräteverhalten

VA EinAus	VAIndivi- dualverkehr EinAus	ÖPNV EinAus	Geräteverhalten
Aus	Aus	Aus	Übergeordneter VA-Zustand ist aus, d.h. Gerät läuft in Festzeit, keine VA aktiv und damit keine Beeinflussung möglich
Aus	Aus	Ein	Übergeordneter VA-Zustand ist aus, d.h. Gerät läuft in Festzeit, keine VA aktiv und damit keine Beeinflussung möglich
Aus	Ein	Aus	Übergeordneter VA-Zustand ist aus, d.h. Gerät läuft in Festzeit, keine VA aktiv und damit keine Beeinflussung möglich
Aus	Ein	Ein	Übergeordneter VA-Zustand ist aus, d.h. Gerät läuft in Festzeit, keine VA aktiv und damit keine Beeinflussung möglich
Ein	Aus	Aus	VA läuft im Festzeitbetrieb ohne Beschleunigung des IV / ÖPNV, d.h. nur Hintergrundfunktionen der VA (z.B. Archive schreiben, sonstige Funktionen) werden ausgeführt.
Ein	Aus	Ein	VA läuft verkehrsabhängig ohne Beschleunigung des IV, d.h. nur der ÖPNV beeinflusst die Signalisierung.
Ein	Ein	Aus	VA läuft verkehrsabhängig ohne Beschleunigung des ÖPNV, d.h. nur der Individualverkehr beeinflusst die Signalisierung.
Ein	Ein	Ein	VA läuft voll verkehrsabhängig, d.h. sowohl der Individualverkehr als auch der ÖPNV beeinflusst die Signalisierung.
keiner	Aus	Aus	undefinierter Zustand
keiner	Aus	Ein	undefinierter Zustand
keiner	Ein	Aus	undefinierter Zustand
keiner	Ein	Ein	undefinierter Zustand

3.4.9 Projektspezifische Modifikationen

Für die Bedienung und Anzeige von projektspezifischen Erweiterungen die der Gerätehersteller im Gerät implementiert, wie beispielsweise das Schalten von digitalen Ausgängen, sind die Objekte ZProjEinAus, IProjEinAus vorgesehen. Damit ist eine OCIT-Zentrale in der Lage diese zu schalten (ZProjEinAus) und anzuzeigen (IProjEinAus). Damit die Zentrale die Bedeutung einer Modifikation dem Bediener anzeigen kann liefert das Gerät einen Bedeutungstext.

Solange im Gerät eine projektspezifische Erweiterung nicht implementiert ist:

- liefern die Methoden ZProjEinAus.Schalte(..), ZProjEinAus.Get(), IProjEinAus.Get() einen Fehler zurück (ERR_TYPE oder NOT_CONFIGURED),
- wird ZProjEinAus weder im IstVektor.Modifikationen[] noch im ZentralenSchaltwunsch.Get(..Modifikationen[]) übertragen.

Damit auch mehrere gleichartige projektspezifische Modifikationen machbar sind erweitern die Objekte ZProjEinAus, IProjEinAus den Path ihre Basisklassen ZModEinAus, IModEinAus um eine Nummer (ProjModNr) zur Unterscheidung.

3.4.9.1 Objekt ZProjEinAus

Von der Zentrale einstellbare Ein-Ausschaltzustände der projektspezifischen Modifikationen.

ZProjEinAus (1:234)

Objekt ZProjEinAus		
Path	RelKnotenNr	Knotennummer innerhalb eines Lichtsignalsteuergerätes, (von Basisklasse ZModEinAus geerbt).
	ProjModNr	Nummer zur Unterscheidung mehrerer projektspezifischer Modifikationen innerhalb eines Knotens.
METHOD	Name	Beschreibung
16	Schalte	Nächsten Modifikationen Ein- Ausschaltwunsch der Zentrale entgegennehmen. s. ZModEinAus
	Eingabeparameter	
	Vorgang : SYSJOBID	Bedienvorgangskennung des Aufrufers für diese Signalprogrammumschaltung.
	StartZeit: utc	Ab „StartZeit“ gilt dieser Ein- Ausschaltwunsch
	EndZeit : utc	Bis „EndZeit“ gilt dieser Ein- Ausschaltwunsch
	Zustand : ModeEinAusZustand	Einzustellender Zustand siehe ModeEinAusZustand
	Ausgabeparameter	
	RetCode	OK: Auftrag angenommen INTERVALL_INVALID: ungültiges oder verstrichenes Zeitintervall angegeben, Auftrag abgelehnt. PARAM_INVALID: ungültiger VAZustand, Auftrag abgelehnt. NOT_CONFIGURED: Angegebene Modifikation ist nicht versorgt, Auftrag abgelehnt.
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen

Objekt ZProjEinAus		
Path	RelKnotenNr	Knotennummer innerhalb eines Lichtsignalsteuergerätes, (von Basisklasse ZModEinAus geerbt).
	ProjModNr	Nummer zur Unterscheidung mehrerer projektspezifischer Modifikationen innerhalb eines Knotens.
METHOD	Name	Beschreibung
	Aktuell.Vorgang	Bedienvorgangskennung des Aufrufers für diese Modifikationsschaltung
	Aktuell.StartZeit	Ab diesem Zeitpunkt war dieser Schaltwunsch gültig
	Aktuell.EndZeit	Bis zu diesem Zeitpunkt ist oder war dieser Schaltwunsch gültig
	Aktuell.Zustand	Eingestellter Zustand s. ModeEinAusZustand
	Next.Vorgang	
	Next.StartZeit	
	Next.EndZeit	
	Next.Zustand	

3.4.9.2 Objekt IProjEinAus

Aktuell eingestellter Zustand der projektspezifischen Modifikation ProjNr und zugehörige Vorgangskennung.

IProjEinAus (1:235)

Objekt IProjEinAus		
Path	RelKnotenNr	Knotennummer innerhalb eines Lichtsignalsteuergerätes, (von Basisklasse ZModEinAus geerbt).
	ProjModNr	Nummer zur Unterscheidung mehrerer projektspezifischer Modifikationen innerhalb eines Knotens.
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden Ein- Ausschaltung der Modifikation geführt hat.
	Zustand	Ein- Auszustand dieser Modifikation s. ModeEinAusZustand
33	Bedeutung	Zur Abfrage eines Bedeutungstextes zur Anzeige und Unterscheidung in der Zentrale.
	Ausgabeparameter	
	RetCode	OK, NOT_CONFIGURED Modifikation ist nicht versorgt.
	Text : STRING	Bedeutungstext zur Anzeige und Unterscheidung der projektspezifischen Modifikation in der Zentrale.

3.4.10 Objekt ZentralenSchaltwunsch

Dieses Objekt enthält Methoden die mehrere Unterobjekte des Zentralenschaltwunschs betreffen. Es dient auch dazu alle Zentralenschaltwünsche mit einem Get Aufruf zu holen.

ZentralenSchaltwunsch (1:220)

ZentralenSchaltwunsch		
METHOD	Name	Beschreibung
16	SchalteSigProgEin	<p>Diese Methode ist eine Abkürzung um das Signalprogramm und Knoten Ein mit nur einem Aufruf zu schalten.</p> <p>Falls das durch Start- und EndZeit angegebene Intervall ungültig oder in der Vergangenheit liegt, kehrt diese Methode mit RetCode = INTERVALL_INVALID zurück.</p> <p>Die Serverfunktion in F prüft ob zur angegebenen SigProgNr ein Signalprogramm versorgt ist. Falls nein kehrt sie mit RetCode=PARAM_INVALID zurück.</p> <p>Nun führt diese Methode sinngemäß die folgenden Operationen aus: RetCode = ZSignalProgramm.Schalte(Vorgang, StartZeit, EndZeit, SigProgNr); if(RetCode == OK) RetCode = ZKnotenEinAus.Schalte(Vorgang, StartZeit, EndZeit, Ein); return RetCode;</p>
	Eingabeparameter	
	Vorgang : SYSJOBID	Bedienvorgangskennung des Aufrufers für diese Signalprogrammumschaltung.
	StartZeit: utc	Ab „StartZeit“ gilt dieser Signalprogrammumschaltwunsch
	EndZeit : utc	Bis „EndZeit“ gilt dieser Signalprogrammumschaltwunsch
	SigProgNr : ui1	0 Freigabe der lokalen Signalprogrammwahl 1-255 Signalprogramme.
	Ausgabeparameter	
	RetCode	OK: Auftrag angenommen INTERVALL_INVALID: ungültiges oder verstrichenes Zeitintervall angegeben, Auftrag abgelehnt. PARAM_INVALID: ungültige Signalprogrammnummer, Auftrag abgelehnt.
17	LoescheStoerungsblockierung	Falls das Gerät durch eine Störung abgeschaltet hat, ermöglicht diese Methode einen erneuten Einschaltversuch. Die Methode kehrt sofort zurück (wartet nicht bis erneut eingeschalten ist).
	Ausgabeparameter	

ZentralenSchaltwunsch		
METHOD	Name	Beschreibung
	RetCode	OK: Auftrag angenommen PARAM_INVALID: es liegt keine Störungsblockierung vor, Auftrag abgelehnt.
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	SignalProgramm : ZSignalProgramm	Siehe Objekt ZSignalProgramm
	KnotenEinAus : ZKnotenEinAus	Siehe Objekt ZKnotenEinAus
	Teilknoten[0..3] : ZTeilknoten	Übertragung als Array mit festem Typ: - zuerst ein UBYTE Anzahl folgender ZTeilknoten Daten - Daten der Teilknoten (siehe Objekt ZTeilKnoten) Die Daten der Teilknoten werden in aufsteigender Reihenfolge übertragen (keine Übertragung der Teilknotennummer).
	SonderEingriff : ZSonderEingriff	Siehe Objekt ZSondereingriff
Modifikationen[0..15] : ZModEinAus	Modifikationen des Signalprogramms. Hier kann jede von ZmodEinAus abgeleitete Klasse stehen, dies sind z. Zt. ZVAEinAus, ZOepnvEinAus, ZProjEinAus. Übertragung als Array mit variablen Typen: - zuerst ein UBYTE Anzahl folgender ZModEinAus Daten - RefLen, Länge der Referenz - ID der Daten, OdgMember OType - relativer Path (im Fall von ZVAEinAus, ZOepnvEinAus nichts, bei ZProjEinAus ist es ProjModNr) - Datenlänge - Daten einer von ZModEinAus abgeleiteten Klasse	
18	SchalteKnoten	Diese Methode ist eine Abkürzung um das Signalprogramm, Knoten, Teilknoten, Sondereingriff und Modifikationen mit nur einem Aufruf zu schalten. Falls das durch Start- und EndZeit angegebene Intervall ungültig oder in der Vergangenheit liegt, kehrt diese Methode mit RetCode = INTERVALL_INVALID zurück.
	Eingabeparameter	
	Vorgang : SYSJOBID	Bedienvorgangskennung des Aufrufers für diese Schaltung.
	StartZeit: utc	Ab „StartZeit“ gilt dieser Schaltwunsch
	EndZeit : utc	Bis „EndZeit“ gilt dieser Schaltwunsch
	SigProgNr : ui1	0 Freigabe der lokalen Signalprogrammwahl 1-255 Signalprogramme.
	KZustand : ui1	Siehe KZustand

ZentralenSchaltwunsch		
METHOD	Name	Beschreibung
	Teilknoten[0..3] : TKZustand	Übertragung als Array mit festem Typ: - zuerst ein UBYTE Anzahl folgender TKZustand Daten - Daten der Teilknoten (siehe Objekt TKZustand) Die Daten der Teilknoten werden in aufsteigender Reihenfolge übertragen (keine Übertragung der Teilknotennummer).
	SonderEingriff : ui1	Nummer eines Sondereingriffs.
	Modifikationen[0..15]: GMod-Zustand	Modifikationen des Signalprogramms. Hier kann jede von GModZustand abgeleitete Klasse stehen, dies sind z.Zt. GVAZustand, GOepnvZustand, GVAIndividualverkehrZustand, GProjZustand. Übertragung als Array mit variablen Typen: - zuerst ein UBYTE Anzahl folgender GModZustand Daten - RefLen, Länge der Referenz - ID der Daten, OdgMember OType - relativer Path (im Fall von GVAZustand, GOepnvZustand, GVAIndividualverkehr nichts, bei GProjZustand, ist es ProjModNr) - Datenlänge:ui2 - Modifikationszustand und weitere Daten entsprechend dem zur Laufzeit angegebenen Member, Otype
	Ausgabeparameter	
	RetCode	OK: Auftrag angenommen INTERVALL_INVALID: ungültiges oder verstrichenes Zeitintervall angegeben, Auftrag abgelehnt. PARAM_INVALID: ungültige Signalprogrammnummer, ungültige TK - Zustände, ungültige Sonder-Signalprogrammnummer, ungültige Modifikationszustände angegeben, Auftrag abgelehnt. PATH_INVALID: keine oder ungültige Teilknotennummer angegeben, Auftrag abgelehnt. NOT_CONFIGURED: Angegebene Modifikationen sind nicht versorgt, Auftrag abgelehnt.

Hinweis: Es wird empfohlen ab OCIT-O Lstg Version 2.0, Ausgabe 02 die Methode **SchaltteKnoten** (Methode 18) zu verwenden, weil damit ein Einfluss des zeitlichen Verhaltens der Übertragungsstrecke auf die Schaltwünsche ausgeschlossen wird.

3.4.11 Objekt ISignalProgramm

Das Objekt ISignalProgramm liefert das zum Zeitpunkt der Abfrage vom Gerät bearbeitete Signalprogramm und Vorgangskennung des zugehörigen Auftrags.

Hinweis: Im Falle von vollständigen Störabschaltungen werden die Zustände zum Zeitpunkt der Abschaltung im IstVektor beibehalten.

ISignalProgramm (1:223)

ISignalProgramm		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden SigProgNr Schaltung geführt hat.
	SigProgNr	Nummer des zur Zeit abgearbeiteten Signalprogramms.

3.4.12 Objekt IKnotenEinAus

IKnotenEinAus liefert den zum Zeitpunkt der Abfrage eingestellten Ein- Auszustand des Knotens mit der Vorgangskennung des zugehörigen Auftrags.

Hinweis: Im Falle von vollständigen Störabschaltungen wird der anliegende Auszustand in IstVektor eingetragen.

IKnotenEinAus (1:225)

ISignalProgramm		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden Knoten Ein-Ausschaltung geführt hat.
	KZustand	Ein- Auszustand des Knotens.

3.4.13 Objekt ITeilknoten

Vom Typ ITeilknoten gibt es pro Teilknoten eine Instanz. Um Methoden einer Instanz ITeilKnoten aufzurufen ist im Feld Path (s. Dok.2) die gewünschte Teilknotennummer als ui1 anzugeben.

ITeilknoten enthält den aktuell eingestellten Zustand des adressierten Teilknotens.

Hinweis: Im Falle von vollständigen Störabschaltungen wird der anliegende Auszustand in IstVektor eingetragen.

ITeilknoten (1:227)

ITeilknoten		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden TeilKnoten Ein-Ausschaltung geführt hat.
	Zustand	Ein- Auszustand dieses TeilKnotens.

3.4.14 Objekt ISondereingriff

Das Objekt ISondereingriff liefert das zum Zeitpunkt der Abfrage eingestellte Sonder- Signalprogramm und Vorgangskennung des zugehörigen Auftrags.

Hinweis: Im Falle von vollständigen Störabschaltungen werden die Zustände zum Zeitpunkt der Abschaltung im IstVektor beibehalten.

ISondereingriff (1:229)

ISonderEingriff		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden Sondereingriff Schaltung geführt hat.
	SonderEingriffNr	Nummer des Sondereingriffs. Der Wert 0 bedeutet hier: aktuell kein Sondereingriff.

3.4.15 Objekt IVAEinAus

IVAEinAus ist eine Spezialisierung von IModEinAus und liefert den zum Zeitpunkt der Abfrage eingestellten übergeordneten Zustand der lokalen verkehrsabhängigen Logik des Knotens mit der Vorgangskennung des zugehörigen Auftrags.

Hinweis: Im Falle von vollständigen Störabschaltungen werden die Zustände zum Zeitpunkt der Abschaltung im IstVektor beibehalten.

IVAEinAus (1:231)

IVAEinAus		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden Ein- Ausschaltung der Verkehrsabhängigkeit geführt hat.
	VAZustand	Ein- Auszustand der Verkehrsabhängigkeit.

3.4.16 Objekt IVAIndividualverkehrEinAus

IVAIndividualverkehrEinAus ist eine Spezialisierung von IModEinAus. Das Objekt liefert den zum Zeitpunkt der Abfrage aktiven Zustand der Beeinflussung der lokalen verkehrsabhängigen Logik des Knotens durch den Individualverkehr mit der Vorgangskennung des zugehörigen Auftrags.

Hinweis: Im Falle von vollständigen Störabschaltungen werden die Zustände zum Zeitpunkt der Abschaltung im IstVektor beibehalten.

IVAIndividualverkehrEinAus (1:239)

IVAIndividualverkehrEinAus		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden Ein- Ausschaltung der Verkehrsabhängigkeit geführt hat.
	VAModifikationZustand	Ein- Auszustand des Einflusses des Individualverkehrs auf die VA (Steuerung der VA-Reduzierung)

3.4.17 Objekt IOepnvEinAus

IOepnvEinAus ist eine Spezialisierung von IModEinAus und liefert den zum Zeitpunkt der Abfrage eingestellten übergeordneten Zustand der lokalen ÖPNV Bevorzugung des Knotens mit der Vorgangskennung des zugehörigen Auftrags.

Hinweis: Im Falle von vollständigen Störabschaltungen werden die Zustände zum Zeitpunkt der Abschaltung im IstVektor beibehalten.

IOepnvEinAus (1:233)

IOepnvEinAus		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur folgenden Ein- Ausschaltung der lokalen ÖPNV Bevorzugung geführt hat.
	VAZustand	Ein- Auszustand der lokalen ÖPNV Bevorzugung.

3.4.18 Objekt IBetriebsart

Das Objekt IBetriebsart gibt Auskunft über die aktuell laufende Betriebsart (mit Vorgangsnummer) eines relativen Knotens.

Hinweis: Es gibt keine OCIT-Outstations Funktion um die Betriebsart von der Zentrale aus einzustellen. Eine Betriebsart ist gültig, wenn sie mindestens über einen Zustand die Kontrolle ausübt. Die Betriebsart ergibt sich aus dem IstVektor und kann in wenigen Sonderfällen (z. B. Teilknotenblockierung durch Bedienteil oder Schalter) geräteabhängig verschieden sein.

Im Falle von vollständigen Störabschaltungen werden die Zustände zum Zeitpunkt der Abschaltung im IstVektor beibehalten.

IBetriebsart (1:209)

IBetriebsart (1:209)		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Vorgang	Vorgangskennung des Vorgangs welcher zur Einstellung der Betriebsart geführt hat

IBetriebsart (1:209)		
METHOD	Name	Beschreibung
	Betriebsart	Eingestellte Betriebsart: <ul style="list-style-type: none"> - Sonderbetrieb - Eigensteuerung - Handstoppbetrieb - LokalFix - LokalZeitsteuerung - Zentrale

3.4.19 Objekt IstVektor

Das Objekt IstVektor liefert den aktuellen Betriebszustand und eine Sammelstörungskennung. Wenn sich die Sammelstörung ändert, wird dazu ein Event-Telegramm abgesetzt. Daraufhin kann die Zentrale den IstVektor lesen.

Objekt IstVektor (1:221)

Objekt IstVektor (1:221)			
METHOD	Name	Beschreibung	
0	Get		
	Ausgabeparameter		
	RetCode	OK: folgende Parameter richtig gelesen	
	Zeitstempel : utc	Zeitstempel des Gerätes: wann der Zustand erreicht oder zuletzt geändert wurde.	
	Sammelstörung : ui1	0= keine Störung 1= Störung ohne Abschaltung 2= Störung mit Abschaltung 3= Störung mit Teilabschaltung 4= Interne Störung ohne Abschaltung Weitere Festlegungen dazu am Ende der Tabelle!	
	IBetriebsart	Zeigt an, welche Betriebsart aufgrund welchen Vorgangs eingestellt ist.	
		Vorgangskennung	Vorgangskennung des Systemteils welcher die Kontrolle über die aktuell eingestellten Betriebsart ausübt.
		Betriebsart	Zu dieser Zeit eingestellte Betriebsart.
	ISignalProgramm	Zeigt an welches Signalprogramm aufgrund welches Vorgangs eingestellt ist.	
		Vorgangskennung	Vorgangskennung des Systemteils welcher die Kontrolle über das aktuell eingestellten Signalprogramm ausübt.
		SigProgNr	Zu dieser Zeit eingestellte Signalprogrammnummer
		IKnotenEinAus	Zeigt an ob Gesamtknoten ein oder ausgeschaltet ist.

Objekt IstVektor (1:221)

METHOD	Name	Beschreibung
	Vorgangskennung	Kennung des Vorgangs, welcher zu folgendem KZustand führte.
	KZustand	Siehe KZustand.
	ITeilknoten[]	Übertragung aller vorhandenen ITeilknoten als Array mit festen Typen: - Anzahl: UBYTE folgender ITeilknotenstrukturen - relativer Path = Teilknotennummer
	Vorgangskennung	Vorgangskennung des Systemteils welcher die Kontrolle über den aktuell eingestellten TKZustand ausübt.
	TKZustand	Zu dieser Zeit eingestellter Teilknotenstatus. Siehe TKZustand.
	ISondereingriff	Zeigt die aktuell eingestellte Einsatzfahrzeugroute.
	Vorgangskennung	Vorgangskennung des Systemteils welcher die Kontrolle über den aktuell eingestellten Sondereingriff ausübt.
	SondereingriffNr	Siehe SonderEingriffNr.
	Modifikationen[0..15]	Modifikationen des Signalprogramms. Hier kann jede von IModEinAus abgeleitete Klasse stehen, dies sind z. Zt. Objekt IVAEinAus, IOepnvEinAus, IProjEinAus. Übertragung als Array mit variablen Typen: - zuerst ein UBYTE Anzahl folgender IModEinAus Daten - RefLen, Länge der Referenz - OdgMember Otype - relativer Path = ProjModNr nur falls IProjEinAus folgt - Datenlänge - Daten einer von IModEinAus abgeleiteten Klasse
	Vorgangskennung	Vorgangskennung des Systemteils welcher die Kontrolle über den aktuell eingestellten Zustand der Modifikation ausübt.
	Zustand	Zustand der Modifikation

Festlegung zur Kennung Sammelstörung:

Das Sammelstörungsbyte hat fest definierte Werte für die verschiedenen Fehlerkategorien, so dass bei mehreren gleichzeitig vorliegenden Fehlerarten eine Priorisierung vorgenommen werden muss, d.h. ein Fehler mit einer höheren Priorität übersteuert einen Fehler niedriger Priorität. Die Sammelstörung wird gesetzt wie folgt:

- Störung ohne Abschaltung (1) (Priorität 2):
 - sekundäre Lampenfehler (ohne Abschaltung)
 - sonstige Signalsicherungs-Alarme (Sisi-Alarme) ohne Abschaltung
- Störung mit Abschaltung der gesamten Anlage (2) (Priorität 4):
 - Netzausfall (nur möglich bei USV)
 - alle Sisi-Störabschaltungen (z.B. primäre Lampenfehler)
 - Abschaltung wg. Umlaufkontrolle, gravierende interne Fehler (z.B. nicht auflösbare Feindlichkeit im Signalplan...) welche zu einer Abschaltung führen.
- Störung mit Teilabschaltung der Anlage (3) (Priorität 3):
 - Abschaltung von Teilknoten durch Sisi, aber mindestens 1 Teilknoten läuft noch
 - Abschaltung eines Teilknotens wegen einem internen Fehler
- Interne Störung ohne Abschaltung (4) (Priorität 1):
 - Kommunikationsstörungen
 - Detektorstörung
 - Störung ÖV – Empfang
 - Ortsbetrieb / Festzeitbetrieb als Rückfallebene (z.B. VA abgeschaltet wg. Fehler, Umlaufkontrolle...)
 - ein wichtiger Systemprozess hat einen Fehler (z.B. (Teil-)Prozess reagiert nicht mehr)
 - Zeitquelle gestört

3.4.20 Objekt Gerätestatus

Neben dem Istvektor gibt es einen Gerätestatus pro Lichtsignalsteuergerät. Dieser ist abfragbar, wird aber nicht in das Betriebszustandsarchiv geschrieben, da er relativ umfangreich ist, und das sich oft ändernde Betriebszustandsarchiv unnötig vergrößern würde. Beim Auftreten

von Störungen generiert das Lichtsignalsteuergerät entsprechende Meldungen im Meldungsarchiv.

Geraetestatus (1:236)

Geraetestatus		
METHOD	Name	Beschreibung
0	Get	
	Ausgabeparameter	
	RetCode	OK: folgende Parameter richtig gelesen
	Zeitquelle : ui1	Gibt die aktuelle Quelle der Gerätezeit an
	NotAus : bool	Gibt an, ob ein ggf. vorhandener NOTAUS Schalter betätigt ist
	TuerAuf : bool	TuerAuf=true bedeutet: Der Türschließkontakt meldet: Mindestens eine Tür des Gerätes ist offen. Wenn kein Türschließkontakt vorhanden ist, ist TuerAuf=false
	NetzSpannungOk : bool	Gibt an, ob die für den vollen Gerätebetrieb nötige Netzspannung anliegt
	Gestörte Detektoren	Liste der gestörten Detektoren
	Gestörte Lampen	Liste der gestörten Lampen
	PersistenzSpeicherOk : bool	Gibt an, ob der gesamte Persistenzspeicher konsistent ist. Dieses Flag wird nach Netz Ein oder häufiger vom Gerät gesetzt

3.5 Meldungen und Messwerte

In Archiven der Lichtsignalsteuergeräte werden ausgewählte Betriebsdaten gesammelt. In jedem Gerät existieren mehrere Archive. Welche Daten in welchem Archiv gespeichert werden wird durch Aufträge der Zentrale festgelegt. Pro Archiv sind bis zu 256 verschiedene Aufträge möglich. OCIT-Outstations vereinigt die bisher getrennten Messwert- und Meldungsarchive unter einer gemeinsamen Schnittstelle. Die Datenstrukturen und die definierten Funktionen der Schnittstelle sind für Meldungen und Messwerte strukturell gleich.

Die Daten aus den Archiven können von der Zentrale oder über Tools am Systemzugang ausgelesen werden. Dazu kann die Zentrale von Gerät archivierte Daten die an bestimmten Positionen stehen oder Daten die zu bestimmten Zeiten erfasst wurden anfordern. Im Normalbetrieb werden die archivierten Daten von der Zentrale beim Eintreten bestimmter Ereignisse abgeholt. Beim Eintritt eines solchen Ereignisses sendet das Gerät ein Event-Telegramm (enthält nicht die Daten) an die Zentrale, die daraufhin einzelne oder mehrerer Daten aus den Archiven anfordern kann. Event-Telegramme können ausgelöst werden:

- bei Erreichen eines eingestellten Füllgrads des Archivs,
- beim Eintragen bestimmter variabler Werte,
- bei Änderung der Zieladresse für die Event-Telegramme.

Die Archive der Geräte können während des Betriebs über die Zentrale parametrierbar werden. Festgelegt werden können: Größe, Art der Aufträge, Ereignisse die zu Event-Telegrammen führen, Erfassung von Daten Anhalten und Freigeben, Reset.

Eine ausführliche Beschreibung der Handhabung von Meldungen und Messwerten findet sich im Dokument OCIT-O-Basis.

Die für Lichtsignalsteuergeräte definierten Archive sind in Pkt. 3.5.6 beschrieben.

3.5.1 Objekttypen und Klassenübersicht

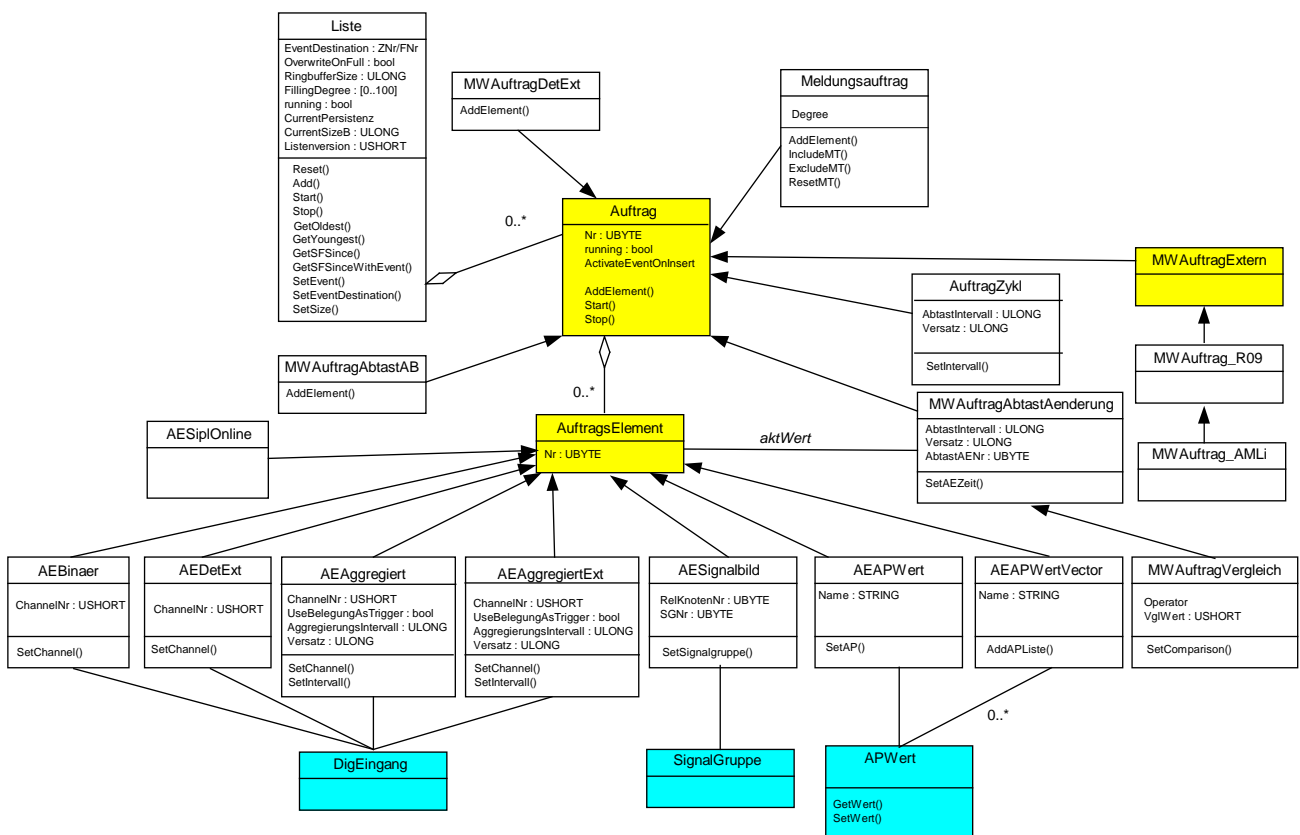


Bild 6: Meldungen und Messwerte: Schema der Objekttypen und Klassen (Auswahl)

Member=1:OType

OType	Name	Pfad (ab Lichtsignalsteuergerät)
403	AuftragZykl	Liste(UBYTE)/Auftragsnummer(UBYTE)
406	MWAuftragAbtastAB	Liste(UBYTE)/Auftragsnummer(UBYTE)
407	MWAuftragAbtastAenderung	Liste(UBYTE)/Auftragsnummer(UBYTE)
408	MWAuftragVergleich	Liste(UBYTE)/Auftragsnummer(UBYTE)
409	MWAuftragExtern	Liste(UBYTE)/Auftragsnummer(UBYTE)
410	MWAuftragR09	Liste(UBYTE)/Auftragsnummer(UBYTE)
411	MWAuftragAMLi	Liste(UBYTE)/Auftragsnummer(UBYTE)
412	MWAuftragDetExt	Liste(UBYTE)/Auftragsnummer(UBYTE)
431	AEBinaer	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
432	AEAggregiert	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
433	AESignalBild	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
434	AEAPWert	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
435	AEDetExt	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
436	AEAggregiertExt	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
437	AEAPWertVektor	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
438	AEsiplOnline	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
439	AEDigAusgang	Liste(UBYTE)/Auftragsnummer(UBYTE)/AENr(UBYTE)
500	DigEingang	ChannelNr(USHORT)
501	SignalGruppe	RelKnotenNr(UBYTE)/SignalgruppenNr(UBYTE)
502	SignalGeber	RelKnotenNr(UBYTE)/DigAusgangNr(USHORT)
503	SignalKammer	RelKnotenNr(UBYTE)/SignalKammerNr(USHORT)
504	DigAusgang	RelKnotenNr(UBYTE)/DigAusgangNr(USHORT)
505	APWert	Name(STRING)
506	APWertUShort	Name(STRING)
507	APWertLong	Name(STRING)
508	APWertBlock	Name(STRING)
510	APWertRk	Name(STRING)/ RelKnotenNr(UBYTE)
511	APWertRkUShort	Name(STRING)/ RelKnotenNr(UBYTE)
512	APWertRkLong	Name(STRING)/ RelKnotenNr(UBYTE)
513	APWertRkBlock	Name(STRING)/ RelKnotenNr(UBYTE)
515	APWertGroup	Name(STRING)
516	APWertGroupRk	Name(STRING)/ RelKnotenNr(UBYTE)

Alle Objekte mit Ausnahme des Events unterstützen die Standardfunktion ‚Get‘. Sie unterstützen nicht die Funktion ‚Set‘. Die zurückgelieferten Parameter werden in der XML-Datei genauer beschrieben.

3.5.2 Messwertaufträge für Lichtsignalanlagen

Siehe auch Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis.

3.5.2.1 Zyklisch abgefragter Auftrag (AuftragZykl)

Der zyklisch abgefragte Auftrag (AuftragZykl) trägt die Auftrags Elemente zyklisch ein. Die Zeitpunkte TZykl und Versatz werden im Sekunden-Maßstab eingetragen, sie beziehen sich auf das eingestellte Rückrechenverfahren (siehe Pkt. 2.5)

AuftragZykl ist abgeleitet von Auftrag. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

AuftragZykl (1:403)

Zyklisch abgefragter Auftrag (AuftragZykl)		
METHOD	Name	Beschreibung
120, 121, 122	AddElement, Start, Stopp	Siehe Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis.
130	SetZyklus	setzt die Zykluszeit und den Versatz im Sekundenraster.
	Eingabeparameter	
	Abtastintervall : ULONG	Zykluszeit in 10 Millisekunden-Einheiten.
	Versatz : ULONG	Signalzeiten Versatz gegenüber den Standard „OCIT-Outstations-Rückrechenverfahren“ (Pkt. 2.5) in 10 Millisekunden-Einheiten. Der Versatz wird MOD Abtastintervall gerechnet.
	Ausgabeparameter	
	RetCode	OK: wird zurückgeliefert, wenn der Zyklus erfolgreich gesetzt wurde. CYCLE_TOO_SHORT : Die Zykluszeit ist zu kurz.
	MinIntervall : ULONG	Zykluszeit >= übergebene Abtastintervall, in dem das Gerät abtasten kann.
	ListenversionAlt, ListenversionNeu	Listenversion vor SetZyklus Listenversion nach SetZyklus

3.5.2.2 Auftrag bei Abtaständerungen (MWAuftragAbtastAenderung)

Auftrag bei Abtaständerungen. Dieser Auftrag betrachtet den Wert eines Auftragslements (eine Prozessvariable) im angegebenen Intervall. Ändert sich dieser Wert, so schreibt der Auftrag einen Sekundenframe.

MWAuftragAbtastAenderung ist abgeleitet von Auftrag. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

MWAuftragAbtastAenderung (1:407)

Abtaständerung (MWAuftragAbtastAenderung)		
METHOD	Name	Beschreibung
120, 121, 122	AddElement, Start, Stopp	Siehe Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis.
130	SetAEZeit	Setzt die Zykluszeit und den Versatz im Sekundenraster.
	Eingabeparameter	
	AbtastAENr : UBYTE	Nummer des Auftragslements, das abgetastet wird.
	AbtastIntervall : ULONG	Zeitraster in 10ms, in denen abgetastet wird.
	Versatz : ULONG	Versatz gegenüber den Standard-OCIT-Outstations-Rückrechenverfahren in 10 Millisekunden-Einheiten. Der Versatz wird MOD Abtastintervall gerechnet.
	Ausgabeparameter	
	RetCode	OK: wird zurückgeliefert, wenn der Zyklus erfolgreich gesetzt wurde. CYCLE_TOO_SHORT : Die Zykluszeit ist zu kurz.
	MinIntervall: ULONG	Zykluszeit >= übergebene AbtastIntervall, in dem das Gerät abtasten kann.
119	ActivateEvent	Aktiviert bzw. deaktiviert den EventEvList::OnInsert, der ausgelöst wird, wenn in die Liste ein Element dieses Auftrags kommt.
	Eingabeparameter	
	ActivateIt : UBYTE	0: Der Event wird deaktiviert 1: Der Event wird aktiviert
	Ausgabeparameter	
		OK: Der Event wurde erfolgreich aktiviert bzw. deaktiviert.
	ListenversionAlt, ListenversionNeu	Listenversion vor ActivateEvent Listenversion nach ActivateEvent

Hinweis: Wenn eine Liste mit einem Auftrag MWAuftragAbtastAenderung ohne Konfiguration des Intervalls (SetAEZeit) gestartet wird, dann hat der Auftrag das Intervall 0 und liefert damit keine Einträge in die Liste.

3.5.2.3 Auftrag Abtaständerungen mit Wertevergleich (MWAuftragVergleich)

Dieses Verfahren ist eine Spezialisierung der Prüfung auf Werteänderung. Das Verfahren überträgt nur wenn a) eine Werteänderung stattgefunden hat und b) die eingegebene Bedingung erfüllt ist.

MWAuftragVergleich ist abgeleitet von MWAuftragAbtastAenderung. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

MWAuftragVergleich (1:408)

Abtaständerung mit Wertevergleich (MWAuftragVergleich)		
METHOD	Name	Beschreibung
120, 121, 122	AddElement, Start, Stopp	Siehe Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis.
130	SetAEZeit	Setzt die Zykluszeit und den Versatz im Sekundenraster.
	Eingabeparameter	
	AbtastAENr : UBYTE	Nummer des Auftragslements, das abgetastet wird.
	AbtastIntervall : ULONG	Zeitraster in 10ms, in denen abgetastet wird.
	Versatz : ULONG	Versatz gegenüber den Standard-OCIT-Outstations-Rückrechenverfahren in 10 Millisekunden-Einheiten. Der Versatz wird MOD Abtastintervall gerechnet.
	Ausgabeparameter	
	RetCode	OK: wird zurückgeliefert, wenn der Zyklus erfolgreich gesetzt wurde. CYCLE_TOO_SHORT : Die Zykluszeit ist zu kurz.
	MinIntervall : ULONG	Zykluszeit >= übergebene AbtastIntervall, in dem das Gerät abtasten kann.
150	SetComparison	Setzt die Zykluszeit und den Versatz im Sekundenraster.
	Eingabeparameter	
	Operator : CHAR	,>' der aktuelle Wert ist größer als der Vergleichswert ,<' der aktuelle Wert ist kleiner als der Vergleichswert ,=' der aktuelle Wert ist gleich dem Vergleichswert ,!' der aktuelle Wert ist ungleich dem Vergleichswert ,H' der durch den HS-Operator (s. u.) gebildete Wert ist größer als der Vergleichswert
	Vergleichswert : LONG	Vergleichswert für die Operation.

Abtaständerung mit Wertevergleich (MWAuftragVergleich)		
METHOD	Name	Beschreibung
	Ausgabeparameter	
	RetCode	OK: wird zurückgeliefert, wenn der Zyklus erfolgreich gesetzt wurde. UNKNOWN_OP : Nicht unterstützter Operator
	ListenversionAlt, ListenversionNeu	Listenversion vor SetComparison Listenversion nach SetComparison
119	ActivateEvent	Aktiviert bzw. deaktiviert den Event EvList::OnInsert, der ausgelöst wird, wenn in die Liste ein Element dieses Auftrags kommt.
	Eingabeparameter	
	ActivateIt : UBYTE	0: Der Event wird deaktiviert 1: Der Event wird aktiviert
	Ausgabeparameter	
	RetCode	OK: Der Event wurde erfolgreich aktiviert bzw. deaktiviert.
	ListenversionAlt, ListenversionNeu	Listenversion vor ActivateEvent Listenversion nach ActivateEvent

Der HS-Operator bildet sich wie folgt:

Es wird ein Akkumulator vom Typ LONG benötigt. Bei Eintrag eines Sekundenframe wird der Akkumulator auf 0 gesetzt. Pro Abtastintervall wird die Differenz des aktuellen Werts vom letzten eingetragenen Wert gebildet (inkl. Vorzeichen, also nicht der Betrag!) und auf den Akkumulator addiert. Wenn der Betrag des Akkumulatorwerts größer als der Vergleichswert ist, löst die Comparison aus.

Der HS Operator ist nur für Werte sinnvoll, bei denen (stetige) Werte von einem Rauschen überlagert sind und bei denen signifikante Änderungen interessieren.

3.5.2.4 Auftrag für asynchrone Prozessvariable (MWAuftragExtern)

Das Objekt MWAuftragExtern ist nur dann notwendig, wenn Ereignisse von Zusatzeinrichtungen erfasst werden müssen wie z. B. von einer ÖPNV-Einrichtung.

Auftrag für Prozessvariable die asynchron durch externe Ereignisse entstehen.

MWAuftragExtern (1:409) ist abgeleitet von Auftrag. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

MWAuftragExtern (1:409)

Auftrag für asynchrone Prozessvariable (MWAuftragExtern)		
METHOD	Name	Beschreibung
120, 121, 122	AddElement, Start, Stopp	Siehe Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis.
119	ActivateEvent	Aktiviert bzw. deaktiviert den EventEvList::OnInsert, der ausgelöst wird, wenn in die Liste ein Element dieses Auftrags kommt.
	Eingabeparameter	
	ActivateIt : UBYTE	0: Der Event wird deaktiviert 1: Der Event wird aktiviert
	Ausgabeparameter	
		OK: Der Event wurde erfolgreich aktiviert bzw. deaktiviert.
	ListenversionAlt, ListenversionNeu	Listenversion vor ActivateEvent Listenversion nach ActivateEvent

3.5.2.5 Auftrag für Einzelschleifen-Erfassung (MWAuftragAbtastAB)

Es ist sinnvoll, die digitalen Daten einer Einzelschleife komprimiert im Messwertarchiv abzuliegen. Normalerweise wird bei einem Messwertauftrag die Wertänderung gespeichert, wobei der neue Wert in einem Parameter hinter dem Subsekundeneintrag abgelegt wird.

Da bei binären Signalen der neue Wert sich in einem Bit darstellen lässt und solche Wertänderungen sehr häufig auftreten, wird in diesem Sonderfall das Parameterbyte eingespart und der neue Zustand in **Bit 2⁷ des Subsekundeneintrags im Auftragsframe gespeichert**⁶. Wenn also das Bit $2^7 = 0$ ist, bedeutet das einen Wechsel von „1“ nach „0“; ist es 1 impliziert es einen Wechsel von „0“ nach „1“.

Beim Start dieses Auftrags wird der Anfangszustand dargestellt, indem zum Zeitpunkt des Auftragsstarts ein Wechsel des Signals vorgetäuscht wird. Beginnt das Signal mit 0 wird also ein Wechsel von 1 nach 0 eingetragen; beginnt das Signal mit 1, wird ein Wechsel von 0 nach 1 zu Beginn gespeichert.

MWAuftragAbtastAB ist abgeleitet von Auftrag. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

⁶ Achtung: Dies ist ein Sonderfall mit einer unterschiedlichen Behandlung des Auftragslements AEBinaer! Bei anderen Aufträgen, z.B. AuftragZykl, wird der digitale Zustand in einem zusätzlich Byte gespeichert!

MWAuftragAbtastAB (1:406)

Auftrag für Einzelschleifen-Erfassung (MWAuftragAbtastAB)		
METHOD	Name	Beschreibung
120	AddElement	AddElement darf bei dem Auftragstyp MWAuftragAbtastAB nur einmal (!) mit dem Typ AEBinaer aufgerufen werden (und sonst mit keinem Typ). Sonst liefert AddElement den Fehler NOT_POSSIBLE zurück!
121, 122	Start, Stopp	Siehe Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis.

3.5.2.6 Auftrag für erweiterte Detektorwerte (MWAuftragDetExt)

Einige Schleifendetektoren liefern bei Konfiguration als Doppelschleife neben der allgemein üblichen Belegungs- und Lückeninformation noch weitere Werte, welche in moderneren Steuerungsverfahren verwendet werden können. In der Regel sind dies Geschwindigkeit, Fahrzeugart, Fahrzeuglänge und die Fahrdauer von der ersten bis zur zweiten Schleife. Um diese Informationen in der Zentrale nutzbar zu machen, wird der nachstehende Auftrag definiert. Wenn der Auftrag gesetzt wird, werden die Datensätze für alle Fahrzeuge, welche den betreffenden Detektor überfahren, abgespeichert.

Hinweis: Da diese Daten völlig asynchron entstehen, soll hier sinnvollerweise auch nur ein Auftragsselement pro Auftrag definiert werden.

Für die als Doppelschleife konfigurierten Detektoren mit erweiterter Abfragemöglichkeit wird ein eigenes Auftragsselement eingeführt. Somit sind sie über die Methode GetInstanceInfo() des Systemobjekts Lichtsignalsteuergerät zugänglich. Weiterhin kann nur so im Auftragsselement die Zulässigkeit einer zugewiesenen Kanalnummer überprüft und gegebenenfalls abgelehnt werden, ohne den zugehörigen Auftrag zu kennen.

MWAuftragDetExt ist abgeleitet von Auftrag. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

MWAuftragDetExt (1:412)

Auftrag für erweiterte Detektorwerte (MWAuftragDetExt)		
METHOD	Name	Beschreibung
120	AddElement,	Siehe Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis. Hier ist nur das Auftragsselement AEDetExt zulässig.
121, 122	Start, Stopp	Siehe Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis.
119	ActivateEvent	aktiviert bzw. deaktiviert den Event EvList::OnInsert, der ausgelöst wird, wenn in die Liste ein Element dieses Auftrags kommt.

Auftrag für erweiterte Detektorwerte (MWAuftragDetExt)		
MET HOD	Name	Beschreibung
	Eingabeparameter	
	ActivateIt : UBYTE	0: Der Event wird deaktiviert 1: Der Event wird aktiviert
	Ausgabeparameter	
	RetCode	OK: Der Event wurde erfolgreich aktiviert bzw. deaktiviert.
	ListenversionAlt, ListenversionNeu	Listenversion vor ActivateEvent Listenversion nach ActivateEvent

Auftragselement für Doppelschleifen mit Zusatzinformationen (AEDetExt) siehe Pkt. 3.5.3.5

Auftragselement für erweiterte aggregierte Det.-Werte (AEAggregiertExt), siehe Pkt.3.5.3.2 .

3.5.2.7 Auftrag für R09-Telegramme (MWAuftragR09)

Der R09-Auftrag hat – anders als die anderen Messwert-Aufträge keine Auftragselemente. Stattdessen ist mit dem R09-Auftrag unmittelbar eine Struktur verbunden. Der R09-Auftrag generiert immer Frames des Typs MWAuftragFrameR09. Wenn der Auftrag gesetzt wird, werden alle für dieses Lichtsignalsteuergerät relevanten R09-Telegramme abgespeichert. Irrelevante Telegramme, die trotzdem empfangen wurden, werden nicht gespeichert.

MWAuftragR09 ist abgeleitet von MWAuftragExtern. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

MWAuftragR09 (1:410)

MWAuftragR09		
METHOD	Name	Beschreibung
120, 121, 122	AddElement, Start, Stopp	Siehe Kapitel „Abläufe Meldung und Messwerte“ OCIT-O-Basis.
119	ActivateEvent	Aktiviert bzw. deaktiviert den Event EvList::OnInsert, der ausgelöst wird, wenn in die Liste ein Element dieses Auftrags kommt.
	Eingabeparameter	

MWAuftragR09		
METHOD	Name	Beschreibung
	ActivateIt : UBYTE	0: Der Event wird deaktiviert 1: Der Event wird aktiviert
	Ausgabeparameter	
	RetCode	OK: Der Event wurde erfolgreich aktiviert bzw. deaktiviert.
	ListenversionAlt, ListenversionNeu	Listenversion vor ActivateEvent Listenversion nach ActivateEvent

Struktur des dynamischen Datensatzes

Name	Kurzbez.	Datentyp	Wertebereich	Bemerkungen
Tag (Erstelldatum)	TT	UBYTE	1..31	Erstelldatum / Uhrzeit
Monat (Erstelldatum)	MO	UBYTE	1..12	Erstelldatum / Uhrzeit
Jahr (Erstelldatum)	JJ	UBYTE	0..99	Erstelldatum / Uhrzeit
Stunde (Erstelldatum)	HH	UBYTE	0..23	Erstelldatum / Uhrzeit
Minute (Erstelldatum)	MM	UBYTE	0..59	Erstelldatum / Uhrzeit
Sekunde (Erstelldatum)	SS	UBYTE	0..59	Erstelldatum / Uhrzeit
Meldepunktnummer	MPN	LONG	1-2 ²⁴	5 Zeichen im Telegramm
Liniennummer	LLL	USHORT	0-999	3 Zeichen im Telegramm
Kursnummer	KK	UBYTE	0-99	2 Zeichen im Telegramm
Routennummer	RRR	USHORT	0-999	3 Zeichen im Telegramm
Priorität	P	UBYTE	0-7	1 Zeichen im Telegramm
Zuglänge	Z	UBYTE	0-7	1 Zeichen im Telegramm
Richtung Hand	H	UBYTE	0-3	1 Zeichen im Telegramm Manuelle Anforderung durch den Fahrer (z.B. mittels Schlüsselschalter an der Haltestelle).
Fahrplanabw (Min+Sek)	FAHRP	SHORT (signed short)	-3599 bis 3599	„Fahrplanlage“ Abweichung vom Fahrplan in Sekunden.

Anmerkung: Die Felder TT, MO, JJ, HH, MM, SS beschreiben das Datensatzerstelldatum der externen Einheit, bei dem die lokale Zeit als Einheit verwendet wird.

3.5.2.8 Auftrag für erweiterte R09-Telegramme (ÖPNV-Eintrag AMLi)

Der Auftrag MWAuftragAMLi (1:411) für erweiterte R09-Telegramme ist gegenüber dem Auftrag für R09-Telegramme von der Auswahl her gleich und liefert nur einen erweiterten

Datensatz (z.B.: GNA, GNE, TX...) zurück. Die Funktionen entsprechen exakt denen des Auftrags für R09-Telegramms und werden in 3.5.2.7 aufgeführt.

MWAuftragAMLi ist abgeleitet von MWAuftragR09. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

Die erweiterte Datenstruktur sieht folgendermaßen aus:

Name	Kurzbez.	Datentyp	Wertebereich	Bemerkungen
Tag (Erstelldatum)	TT	UBYTE	1..31	Erstelldatum / Uhrzeit
Monat (Erstelldatum)	MO	UBYTE	1..12	Erstelldatum / Uhrzeit
Jahr (Erstelldatum)	JJ	UBYTE	0..99	Erstelldatum / Uhrzeit
Stunde (Erstelldatum)	HH	UBYTE	0..23	Erstelldatum / Uhrzeit
Minute (Erstelldatum)	MM	UBYTE	0..59	Erstelldatum / Uhrzeit
Sekunde (Erstelldatum)	SS	UBYTE	0..59	Erstelldatum / Uhrzeit
Meldepunktnummer	MPN	LONG	1-2 ²⁴	5 Zeichen im Telegramm
Liniennummer	LLL	USHORT	0-999	3 Zeichen im Telegramm
Kursnummer	KK	UBYTE	0-99	2 Zeichen im Telegramm
Routennummer	RRR	USHORT	0-999	3 Zeichen im Telegramm
Priorität	P	UBYTE	0-7	1 Zeichen im Telegramm
Zuglänge	Z	UBYTE	0-7	1 Zeichen im Telegramm
Richtung Hand	H	UBYTE	0-3	1 Zeichen im Telegramm; manuelle Anforderung durch den Fahrer (z.B. mittels Schlüsselschalter an der Haltestelle)
Fahrplanabw (Sek)	FAHRP	SHORT (signed short)	-3599 bis 3599	„Fahrplanlage“ Abweichung vom Fahrplan wie im empfangenen R09 Telegramm.
Relative Knotennummer	RELKN	UBYTE	0..255	Die Nummer des relativen Knotens im Lichtsignalsteuergerät, die ausgewertet wird.
OeV-Modifikation durch Zentrale aktiv?	OEVAKT	UBYTE	0-1	0: OeV Modifikation inaktiv 1: OeV Modifikation aktiv 255: Dieser Wert ist nicht gesetzt
TX bei Meldung	TX	UBYTE	1-255	Umlaufsekunde (Beachte Anmerkungen am Ende der Tabelle!)
Signalplan	SP	UBYTE	0-32	Nummer des aktuellen Signalplans
Laufende Phase	PH	UBYTE	0-255 ?	0: Phase im Verfahren nicht definiert (z.B. VSPlus) 1..255: aktuelle Phasennummer
Gewünschte Phase	UE	UBYTE	0-255	0: Es ist kein Phasenübergang aktiv

Name	Kurzbez.	Datentyp	Wertebereich	Bemerkungen
				bzw. eine Phase ist im Verfahren nicht definiert (VSPlus) 1..255: Es ist ein Phasenübergang aktiv von Phase PH nach Phase UE
Fahrzeit (Bei Abmeldung die echte Fahrzeit von Anmeldung bis Abmeldung. Bei Anmeldung die theoretische, berechnete, Fahrzeit von Anmeldung bis Abmeldung.)	TWF	UBYTE	0-255	0: keine Fahrzeit vorhanden 1..255: Fahrzeit
Bei Abmeldung: Grünanfang der ÖV-Signalgruppe	GNA	UBYTE	0-255	Bezogen auf TX (siehe auch Anmerkungen)
Bei Abmeldung: Grünende der ÖV-Signalgruppe	GNE	UBYTE	0-255	Bezogen auf TX (siehe auch Anmerkungen)

Anmerkungen:

Im Gegensatz zu den OCIT-O konformen 0,1 Sekunden Zeitschaltwerten von $TX = 0$ bis $TX = TU - 1$, gibt es im AMLI-Datensatz aus historischen Gründen nur ganze Sekundenschritte von $TX = 1$ bis $TX = TU$. Die OCIT-O Zeitschaltwerte $TX = n * 0,1$ Sekunden müssen daher in AMLI konforme Werte umgerechnet werden: $OCIT-O TX / 10$ (ohne Rest) + 1.

Beispiele für eine OCIT-O Umlaufzeit von 30,5 Sekunden:

OCIT-O: $TX = 0$ $TX = 10$ $TX = 52$ $TX-1 = 304$

AMLI: $TX=1$ $TX = 2$ $TX = 6$ $TU = 31$

Die TX-Werte in AMLI entsprechen damit der laufenden Sekunde im Umlauf. TU ist die letzte Sekunde des Umlaufs, darauf folgt wieder die erste Sekunde.

Folgende Kennungen werden nur bei Abmeldung gesetzt, bei Anmeldungen sind GNA und GNE immer 0:

Kennung im ÖV-Speicher		Bedeutung
Grün-Anfang	Grün-Ende	
1..253	1..253	Eintrag von TX für Grün der beeinflussten Signalgruppe. Falls die Abmeldung kurz nach der Umschaltung auf Rot (innerhalb der Zeit des VA-Parameters „festgelegte Rotzeit“ ⁷) erfolgt, d. h. dass der Bus bei Gelb / Rot noch gefahren ist, wird das für Grünende bei GNE eingetragen. Erfolgt die Abmeldung später als der parametrierte Wert wird GNE auf 0 gesetzt.
0	0	Die Abmeldung erfolgte nach GNE plus festgelegter Rotzeit der Signalgruppe
0	255	Abmeldung erfolgte im Auszustand des Steuergeräts
254	254	Die SG hatte bei Anmeldung und 15 s ⁸ nach Abmeldung Grün. Die An- und Abmeldung des ÖV hatte keinen Einfluss auf die SG, da die SG im Zustand Dauergrün stand.
254	1..253	Die SG hatte zum Zeitpunkt der Anmeldung schon Grün und blieb bis zur Abmeldung in diesem Zustand. GNE entspricht echtem GNE dieser SG nach Abmeldung des ÖV.
254	0	Die SG hatte zum Zeitpunkt der Anmeldung schon Grün und blieb bis zur Abmeldung in diesem Zustand. Der Datensatz wurde vor GNE der Signalgruppe in den ÖV-Speicher geschrieben, da eine weitere Abmeldung durch einen Folgebus eintraf, bevor die n Sekunden bis GNE gleich 254 abgelaufen waren(siehe auch nächste Kombination).
1..253	254	Grün-Anfang der SG nach der Anmeldung. Die SG hatte 15s nach der Abmeldung immer noch Grün, d.h. GNE der SG wurde nicht durch ÖV-Anforderung beeinflusst. Der Wert von GNA entspricht echtem GNA der SG nach Anmeldung.
1..253	0	Grün-Anfang der SG nach der Anmeldung. Die Abmeldung erfolgte im Grün, ein Folgebus hat noch nicht abgemeldet. Der Datensatz wurde vor GNE der Signalgruppe in den ÖV-Speicher geschrieben, da eine weitere Abmeldung durch einen Folgebus eintraf, bevor die n Sekunden bis GNE gleich 254 abgelaufen waren. Der Wert von GNA entspricht echtem GNA der SG nach Anmeldung.

⁷ „Festgelegte Rotzeit“ ist ein Parameter des VA-Verfahrens oder herstellerspezifisch versorgt.

⁸ Der Wert 15 Sekunden kann herstellerspezifisch / verfahrensspezifisch abweichen.

3.5.3 Auftrags Elemente

Aufträge setzen sich im Normalfall aus Auftrags Elementen zusammen. Wenn ein Auftrag dynamische Daten in einen Sekundenframe schreibt, werden alle Auftrags Elemente ausgelesen und die Daten direkt hintereinander in den Auftragsframe gepackt.

Wie der Auftrag ist auch das Auftrags Element eine „virtuelle Basisklasse“ (mit der virtuellen Methode 150 „GetTriggerValue), d. h. es gibt eine Reihe von speziellen Auftrags Elementen, die instanziiert werden können, aber nicht das Auftrags Element selbst. Jeder Auftrags Elementtyp verweist auf einen bestimmten Typ von Datenquelle, also z.B. auf Digitale Eingänge oder auf Signalgruppen usw. Zudem ist pro Auftrags Element-Typ definiert, welche Daten der Datenquelle bei diesem Auftrags Element-Typ dynamisch abgelegt werden.

- Ein Auftrags Element ist ein OBJTYPE und hat den Pfad:
Liste()/Auftrag()/AuftragsElement()
- Das Auftrags Element definiert die Struktur der Daten im Auftragsframe.
- Im Auftrags Element gibt es eine Referenz auf eine DOMAIN welche den Aufbau der Daten im Auftragsframe beschreibt. (Sinn: Herstellerspezifische Auftrags Elemente können von fremder Zentrale dekodiert werden.)
- Jedes Auftrags Element verweist auf eine Datenquelle. Der Verweis auf die Datenquelle ist typisiert, d.h. ein Auftrags Element eines Typs X verweist immer auf eine Datenquelle des gleichen Typs Y.
- Jedes Auftrags Element liefert auf Anfrage von MWAuftragAbtastungAenderung einen skalaren Wert. Bei strukturierten Auftrags Elementen definiert das Auftrags Element welcher Wert geliefert wird.

3.5.3.1 Auftrags Element für binäre Eingänge (AEBinaer)

Binäre Eingänge (DigEingang, 1:500), wie z.B. Detektoreingänge oder auch Taster werden über das Auftrags Element für binäre Eingänge erfasst. Das Auftrags Element wird in zwei unterschiedlichen Auftragstypen eingesetzt:

- Binäre Eingänge, die sich „selten“ ändern, werden entweder als Auftrags Element in einem beliebigen Auftrag verwendet oder als Trigger in einem (MWAuftragAbtastAenderung bzw. MWAuftragVergleich).
- Binäre Eingänge, die sehr häufig wechseln, wie z.B. die von Schleifendetektoren, werden mit MWAuftragAbtastAB behandelt, der die Daten komprimierter als der normale Auftrag einträgt. Der MWAuftragAbtastAB legt den AEBinaer-Eintrag selbst an.

AEBinaer (1:431)

Auftragsselement für binäre Eingänge (AEBinaer)		
METHOD	Name	Beschreibung
150	GetTriggerValue	liest den Wert, der für die Abtaständerung verwendet wird
	Eingabeparameter	
		Keine
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragsselement einen Triggerwert liefern konnte.
	TriggerValue : LONG	Wert des Triggers: Wert des binären Eingangs (0 oder 1)
151	SetChannel	Setzt die geräteweit eindeutige Channelnummer des binären Eingangs
	Eingabeparameter	
	Channel : DigEingang &	Referenz auf DigEingangsobjekt. ChannelNr ist die verwendete Kanalnummer.
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragsselement hinzugefügt werden konnte PARAM_INVALID : wenn der Kanal nicht existent ist.

Anmerkung: Bei binären Eingängen liefert der aktuelle Triggerwert immer nur die Werte 0 oder 1.

3.5.3.2 Aggregierte Werte für binäre Eingänge (AEAggregiert)

Wenn Schleifendetektoren als binäre Eingänge verwendet werden, ist es ggf. sinnvoll anstelle der Übertragung von Einzelwerten die Zählung und Belegungsgrad bereits im Gerät zu bilden. Die Zählung wird immer normiert in Fz/h (als USHORT) ausgegeben, der Belegungsgrad in % (als UBYTE).

Ein AEAgregiert hat ein Aggregierungsintervall. Es gibt zwei Fälle:

- Ist das Aggregierungsintervall ==0 werden Intervall und Versatz vom zugehörigen Auftrag übernommen.
- Ist das Aggregierungsintervall >0 wird in diesem Zyklus ein neues Aggregierungsintervall begonnen. Das Auftragsselement schreibt immer die Werte des letzten Aggregierungsintervalls in den Sekundenframe. Nur sinnvoll als Element von MWAuftragAbtastÄnderung und MWAuftragAbtastVergleich.

Es werden immer alle beiden Werte (Zählung und Belegungsgrad) in den Frame geschrieben.

AEAggregiert ist abgeleitet von AuftragsElement. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

AEAggregiert (1:432)

Auftragselement für aggregierte binäre Eingänge (AEAggregiert)		
METHOD	Name	Beschreibung
150	GetTriggerValue	Liest den Wert, der für die Abtaständerung verwendet wird.
	Eingabeparameter	
		Keine
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement einen Triggerwert liefern konnte.
	TriggerValue : LONG	Wert des Triggers: Entweder Zählwert oder Belegungsgrad, je nach Wert des Attributs UseBelegungAsTrigger
151	SetChannel	Setzt die geräteweit eindeutige Channelnummer des binären Eingangs
	Eingabeparameter	
	Channel : DigEingang &	Referenz auf DigEingangsobjekt. ChannelNr ist die verwendete Kanalnummer.
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement hinzugefügt werden konnte PARAM_INVALID : wenn der Kanal nicht existent ist.
152	SetIntervall	Setzt das Aggregierungsintervall
	Eingabeparameter	
	AggregierungsIntervall : ULONG	Zeitraaster in 10ms, in denen abgetastet wird.
	Versatz : ULONG	Versatz gegenüber den Standard-OCIT-Outstations-Rückrechenverfahren in 10 Millisekunden-Einheiten. Der Versatz wird MOD AggregierungsIntervall gerechnet.
	Ausgabeparameter	
	RetCode	OK: wird zurückgeliefert, wenn das Aggregierungsintervall erfolgreich gesetzt wurde. CYCLE_TOO_SHORT : Die Zykluszeit ist zu kurz.
	MinIntervall : ULONG	Kleinstmögliches Aggregierungsintervall.

3.5.3.3 Auftragselement für Anwenderprogrammwert (AEAPWert)

Das Auftragselement AEAPWert erfasst Anwenderprogrammwerte (AP-Werte, siehe Pkt. 3.5.4) vom Typ USHORT, ULONG oder BLOB, je nach angegebener AP-Wert Referenz.

AEAPWert ist abgeleitet von AuftragsElement. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

AEAPWert (1:434)

AEAPWert		
METHOD	Name	Beschreibung
150	GetTriggerValue	Liest den Wert, der für die Abtaständerung verwendet wird.
	Eingabeparameter	
		Keine
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement einen Triggerwert liefern konnte.
	TriggerValue : LONG	Wert des Triggers: Wert des APWerts, außer bei APWert-Block, hier wird ein Hashwert verwendet. Hinweis: Der Hashalgorithmus ist vom Gerätehersteller frei wählbar, der Hashwert darf jedoch nicht größer als 32 Bit sein.
153	SetAP	Setzt die Referenz auf den APWert.
	Eingabeparameter	
	APWert : ANYPATH	Referenz auf APWert, Pfad besteht aus: - Referenzlänge, Member, Otype - Name des AP-Werts - weitere Pfadparameter, je nach angegebenen AP-Wert Typ
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement hinzugefügt werden konnte PARAM_INVALID : wenn der APWert nicht existent ist.

3.5.3.4 Auftragselement blockweises Lesen von AP-Werten (AEAPWertVektor)

Die Unterstützung von AEAPWertVektor ist in OCIT-O Lstg V2.0 optional.

AEAPWertVektor ist abgeleitet von AuftragsElement. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

Das Auftragselement AEAPWertVektor erfasst Anwenderprogrammwerte (AP-Werte, siehe Pkt. 3.5.4) vom Typ USHORT, ULONG oder BLOB, je nach angegebener AP Wert Referenz.

Das Auftragselement AEAPWertVektor erlaubt blockweises Lesen von AP-Werten. So können große Mengen von AP-Werten effizienter durch die Zentrale gelesen werden. Der AE-

APWertVektor wird zunächst mit einer Liste von Referenzen auf APWerte initialisiert. Die Werte dieser APWerte werden dann im Block in die Liste geschrieben.

AEAPWertVektor (1:437)

AEAPWertVektor		
METHOD	Name	Beschreibung
156	SetAPListe	Initialisiert das Auftragsselement mit Referenzen auf APWerte. Es gibt die Möglichkeit, der Methode einen Präfix mitzugeben, der den Paths aller Referenzen auf APWerte hinzugefügt wird. Die Maximale Anzahl der APWerte ist 65535 Werte.
	Eingabeparameter	
	Path: Prefix	Der Präfix wird allen Referenzen auf APWerte vorangestellt.
	Path[]: APWerteRefs	APWerteRefs.Anzahl APWerteRefs[].RefLen APWerteRefs[].Member APWerteRefs[].OType APWerteRefs[].... Pfadparameter je nach APWert
	Ausgabeparameter	
	RetCode	OK NOT_INACTIVE falls das Auftragsselement in einem Auftrag ist, der schon gestartet ist. PARAM_INVALID falls einer oder mehrere der Referenzen auf APWerte ungültig sind. In diesem Fall wird keiner der referenzierten APWerte der Auftragsselement zugefügt. TOO_MANY falls das Steuergerät die Menge an APWerten nicht verarbeiten kann.
155	GetAPListe	Liest die mit SetAPListe gesetzte Liste der APWerte.
	Ausgabeparameter	
	Path: Prefix	Der Präfix, der allen Referenzen auf die APWerte vorangestellt wird.
	Path[]: APWerteRefs	APWerteRefs.Anzahl APWerteRefs[].RefLen APWerteRefs[].Member APWerteRefs[].OType APWerteRefs[].... Pfadparameter je nach APWert
150	GetTriggerValue	Liest den Wert, der für die Abtaständerung verwendet wird.
	Ausgabeparameter	

AEAPWertVektor		
METHOD	Name	Beschreibung
	RetCode	OK: wird zurückgeliefert, wenn das Auftragsselement einen Triggerwert liefern konnte.
	TriggerValue: LONG	Wert des Triggers: Hashwert über alle Values der APWerte. Hinweis: Der Hashalgorithmus ist vom Gerätehersteller frei wählbar, der Hashwert darf jedoch nicht größer als 32 Bit sein.

3.5.3.5 Auftragsselement für Detektoren mit Zusatzinformationen (AEDetExt)

AEDetExt ist abgeleitet von AEBinaer. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

AEDetExt (1:435)

Auftragsselement für Detektoren mit Zusatzinformationen (AEDetExt)		
METHOD	Name	Beschreibung
0	Get	
	Eingabeparameter	
		keine
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragsselement hinzugefügt werden konnte.
	AuftragsElementNr	Nummer dieses Auftragslements in seinem Auftrag
	Channel	Referenz auf binären digitalen Eingang.
150	GetTriggerValue	Liest den Wert, der für die Abtaständerung verwendet wird.
	Ausgabeparameter	
	RetCode	NOT_POSSIBLE, weil kein Wert geliefert werden kann.
151	SetChannel	Setzt die geräteweit eindeutige Channelnummer des binären Eingangs
	Eingabeparameter	
	Channel : DigEingang &	Referenz auf DigEingangsobjekt. ChannelNr ist die verwendete Kanalnummer.
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragsselement hinzugefügt werden konnte PARAM_INVALID : wenn der Kanal nicht existent ist.

Struktur des Ergebnisframes (AEDetExtFrame):

Name	Kurzbez.	Datentyp	Wertebereich	Bemerkungen ⁹
Belegung	Bel	USHORT	0..0xFFFE	Belegungsdauer der Messfläche in 10ms; 0 .. 655,34 s
Lücke	Luecke	USHORT	0..0xFFFE	Letzte Lücke in 10ms; 0 .. 655,34 s
Fahrdauer	FD	USHORT	1..0x7FFE	Fahrdauer von der ersten bis zur zweiten Messfläche in ms; 1 ms32766 ms
Geschwindigkeit	GSW	UBYTE	0..0xFE	Gemessene Geschwindigkeit in km/h 0.. 254 km/h
Fahrzeuglänge	FzgLen	UBYTE	1..0xFE	Länge des Fahrzeugs in 0,1m; 0,1..25,4 m
Fahrzeugart	FzgArt	UBYTE	0..0xFF	Art des Fahrzeugs (Klasse) 00h: Pkw 01h: Pkw + Anhänger 02h: Lkw 03h: Lkw + Anhänger 04h: Bus 05h: Sonstige 06h: Motorrad 07h: Lieferwagen 08h: Sattelkraftfahrzeug 09h –FFh: undefiniert

3.5.3.6 Auftragsselement für erweiterte aggr. Det.-Werte (AEAggregiertExt)

Hinweis: Dieses Auftragsselement **AEAggregiertExt** ist eine Erweiterung des Auftragsselements **AEAggregiert**, das nur die im Lichtsignalsteuergerät gebildeten Werte Zählung und Belegungsgrad erfasst. Versionsstand des Lichtsignalsteuergeräts beachten!

AEAggregiertExt ist abgeleitet von AEAggregiert. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

Um Geschwindigkeiten und Fahrzeugarten auch in einer aggregierten Form zu bekommen, wird ein erweitertes aggregiertes Auftragsselement eingeführt. Dieses Auftragsselement **AEAggregiertExt** ist eine Erweiterung des bestehenden Auftragsselements **AEAggregiert**. Es hat auch die gleichen Methoden und wird mit zyklischen Aufträgen verwendet. Die eintreffenden

⁹ Die angegebenen Wertebereiche bezeichnen die mit OCIT-O übertragbaren Informationen. Real übertragene Werte und ihre Genauigkeit sind vom eingesetzten Detektortyp abhängig, der in OCIT-O nicht vorgegeben wird. Geschwindigkeits-Detektoren arbeiten beispielsweise oft erst ab einer Mindestgeschwindigkeit von einigen km/h .

Daten werden gemäß den acht Fahrzeugklassen (entsprechend TLS¹⁰) und „undefiniert“ verteilt, aus denen jeweils die mittlere Geschwindigkeit und ein Zählwert ermittelt werden.

Gibt es keine Werte für die mittlere Geschwindigkeit oder die Zählwerte, so wird der NULLVALUE eingetragen. Somit können auch Detektoren verwendet werden, die nur Teile dieser Informationen liefern.

AEAggregiertExt (1:436)

Auftragselement für erweiterte aggregierte Det. erte ()		
METHOD	Name	Beschreibung
0	Get	
	Eingabeparameter	
		keine
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement hinzugefügt werden konnte.
	AuftragsElementNr	Nummer dieses Auftragselements in seinem Auftrag
	Channel	Referenz auf binären digitalen Eingang.
150	GetTriggerValue	Liest den Wert, der für die Abtaständerung verwendet wird.
	Eingabeparameter	
		Keine
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement einen Triggerwert liefern konnte.
	TriggerValue : LONG	Wert des Triggers: Entweder Zählwert oder Belegungsgrad, je nach Wert des Attributs UseBelegungAsTrigger
151	SetChannel	Setzt die geräteweit eindeutige Channelnummer des binären Eingangs
	Eingabeparameter	
	Channel : DigEingang &	Referenz auf DigEingangsobjekt. ChannelNr ist die verwendete Kanalnummer.
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement hinzugefügt werden konnte PARAM_INVALID : wenn der Kanal nicht existent ist.

Der Datensatz AEAggregiert wird mit dem Frame AEAggregiertExt wie folgt erweitert:

¹⁰ Technische Lieferbedingungen für Streckenstationen, Bundesanstalt für Straßenwesen, 2006

USHORT	Zählwert in Fahrzeuge/h	alle Fahrzeuge im Intervall, wie AEAggregiert
UBYTE	Belegungsgrad in %	alle Fahrzeuge im Intervall, wie AEAggregiert
UBYTE	mittlere Geschwindigkeit Klasse 0 in km/h	0 .. 254; 255 NULLVALUE: kein gültiger Wert
USHORT	Zählwert Klasse 0	0 .. 65534; 65535 NULLVALUE: kein gültiger Wert
UBYTE	mittlere Geschwindigkeit Klasse 1 in km/h	0 .. 254; 255 NULLVALUE: kein gültiger Wert
USHORT	Zählwert Klasse 1	0 .. 65534; 65535 NULLVALUE: kein gültiger Wert
UBYTE	mittlere Geschwindigkeit Klasse 2 in km/h	0 .. 254; 255 NULLVALUE: kein gültiger Wert
USHORT	Zählwert Klasse 2	0 .. 65534; 65535 NULLVALUE: kein gültiger Wert
UBYTE	mittlere Geschwindigkeit Klasse 3 in km/h	(0 .. 254; 255 NULLVALUE: kein gültiger Wert)
USHORT	Zählwert Klasse 3	0 .. 65534; 65535 NULLVALUE: kein gültiger Wert
UBYTE	mittlere Geschwindigkeit Klasse 4 in km/h	0 .. 254; 255 NULLVALUE: kein gültiger Wert
USHORT	Zählwert Klasse 4	0 .. 65534; 65535 NULLVALUE: kein gültiger Wert
UBYTE	mittlere Geschwindigkeit Klasse 5 in km/h	(0 .. 254; 255 NULLVALUE: kein gültiger Wert)
USHORT	Zählwert Klasse 5	(0 .. 65534; 65535 NULLVALUE: kein gültiger Wert)
UBYTE	mittlere Geschwindigkeit Klasse 6 in km/h	(0 .. 254; 255 NULLVALUE: kein gültiger Wert)
USHORT	Zählwert Klasse 6	0 .. 65534; 65535 NULLVALUE: kein gültiger Wert
UBYTE	mittlere Geschwindigkeit Klasse 7 in km/h	0 .. 254; 255 NULLVALUE: kein gültiger Wert
USHORT	Zählwert Klasse 7	0 .. 65534; 65535 NULLVALUE: kein gültiger Wert

UBYTE	mittlere Geschwindigkeit Klasse 8 in km/h	(0 .. 254; 255 NULLVALUE: kein gültiger Wert)
USHORT	Zählwert Klasse 8	0 .. 65534; 65535 NULLVALUE: kein gültiger Wert

3.5.3.7 Auftragselement für Visualisierungsdaten (AESiplOnline)

AESiplOnline		
METHOD	Name	Beschreibung
0	Get	Auslesen der Nummern der Signalgruppen, die zusätzlich zu TX übertragen werden.
	Ausgabeparameter	
	RetCode: USHORT	OK: wird immer zurückgeliefert
	RelKnotenNr: UBYTE	Relative Knotennummer auf die sich dieses Auftragselement bezieht. Hinweis: Default ist RelKnoten 0 (d.h. nach Anlegen des Auftragselements ist die RelKnotenNr mit 0 vorbelegt).
	AnzahlSigru: UBYTE	Anzahl an parametrisierten Signalgruppen
	Sigrunummern[AnzahlSigru] UBYTE	Array mit den Nummern der übergebenen Signalgruppen
150	GetTriggerValue	Liest den Wert, der für die Abtaständerung verwendet wird.
	Eingabeparameter	
		Keine
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement einen Triggerwert liefern konnte.
	TriggerValue: LONG	Wert des Triggers: Hashwert über Tx und alle Signalgruppenzustände. Hinweis: Der Hashalgorithmus ist vom Gerätehersteller frei wählbar, der Hashwert darf jedoch nicht größer als 32 Bit sein.
151	GetSigState	Abfrage des aktuellen TX und Signalzustand
	Ausgabeparameter	
	RetCode: USHORT	OK: wird immer zurückgeliefert

AESiplOnline		
METHOD	Name	Beschreibung
	TX: USHORT	aktuelles TX
	AnzahlSigru: UBYTE	Anzahl an folgenden Signalgruppen
	SigZustand[AnzahlSigru]: UBYTE	Farb-Zustand der einzelnen Signalgruppen (in Signalbildcodierung)
157	SetRelKnotenGetSignal Gruppen	Setzt die relative Knotennummer für dieses Auftragsselement und liefert die Nummern aller Signalgruppen, die zusätzlich zu TX übertragen werden.
	Eingabeparameter	
	RelKnoten: UBYTE	Referenz auf den RelKnoten auf den sich dieses Auftragsselement bezieht.
	Ausgabeparameter	
	RetCode: USHORT	OK: wird zurückgeliefert wenn der RelKnoten gesetzt werden konnte PARAM_INVALID wird zurückgeliefert wenn der RelKnoten nicht gesetzt werden konnte (z.B. wenn er nicht vorhanden ist).
	RelKnotenNr : UBYTE	Relative Knotennummer auf die sich dieses Auftragsselement bezieht.
	AnzahlSigru: UBYTE	Anzahl an parametrisierten Signalgruppen
	Sigrunummern[AnzahlSigru] UBYTE	Array mit den Nummern der übergebenen Signalgruppen

Hinweis: Es wird empfohlen mit Version 2.0 dieses Auftragsselement anstelle AESignalBild zu verwenden.

Eine wichtige Funktion innerhalb einer Zentrale ist die Visualisierung der Signalisierung in Abhängigkeit vom Wert TX. Bisher müssen dafür alle Signalgruppen und der AP-Wert TX als einzelne Auftragsselemente in einem oder mehreren Aufträgen angelegt werden. Zur Vereinfachung wird nun das TX und alle Signalgruppen eines relativen Knotens über ein einzelnes Auftragsselement beauftragt, das in einem Auftragzyklisch oder AbtastAenderung enthalten sein kann.

Es erzeugt als Eintrag im Sekundenframe den Wert TX (USHORT) gefolgt von der Anzahl der Signalgruppen (UBYTE). Darauf folgen die Signalisierungszustände in aufsteigender Reihenfolge von Signalgruppe 1 bis zur angegebenen Anzahl der Signalgruppen. Damit wird die Größe der übertragenen Datenpakete auf das nötigste beschränkt und die Zuordnung der Werte zu den einzelnen Signalgruppen auf Zentralenseite bleibt gewährleistet.

Bei der Beauftragung über den MWAuftragAbtastAenderung wird bei jeder Änderung des TX ein Sekundenframe generiert, der entweder nur das TX enthält (nämlich dann wenn das Signalbild konstant bleibt) bzw. das TX gefolgt von allen Signalgruppen, sobald sich auch nur eine Signalgruppe auf logischer Ebene geändert hat. Bei einem zyklischen Auftrag mit einer Intervalldauer von üblicherweise 1 Sekunde wird jeder Eintrag mit dem TX und allen Signalgruppen des Knotens erzeugt.

Über die Standardmethode Get erhält die Zentrale eine Liste der übertragenen Signalgruppen.

Mit der Methode SetRKGetSignalGruppen muss die Zentrale den gewünschten RelKnoten festlegen und erhält eine Liste der übertragenen Signalgruppen zurück.

Der resultierende Sekundenframe AESiplOnlineFrame des Auftrags ist wie folgt aufgebaut:

USHORT	AP-Wert TX
UBYTE	Anzahl folgender Signalgruppen
UBYTE	Signalisierung[AnzahlSigru]

3.5.3.8 Auftragselement Signalbild (AESignalBild)

Hinweis: Ab Version 2.0 soll dieses Auftragselement nicht mehr verwendet und durch AESiplOnline ersetzt werden.

Das Auftragselement Signalbild referenziert immer logische Signalbilder einer Signalgruppe. AESignalbild ist abgeleitet von AuftragsElement. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

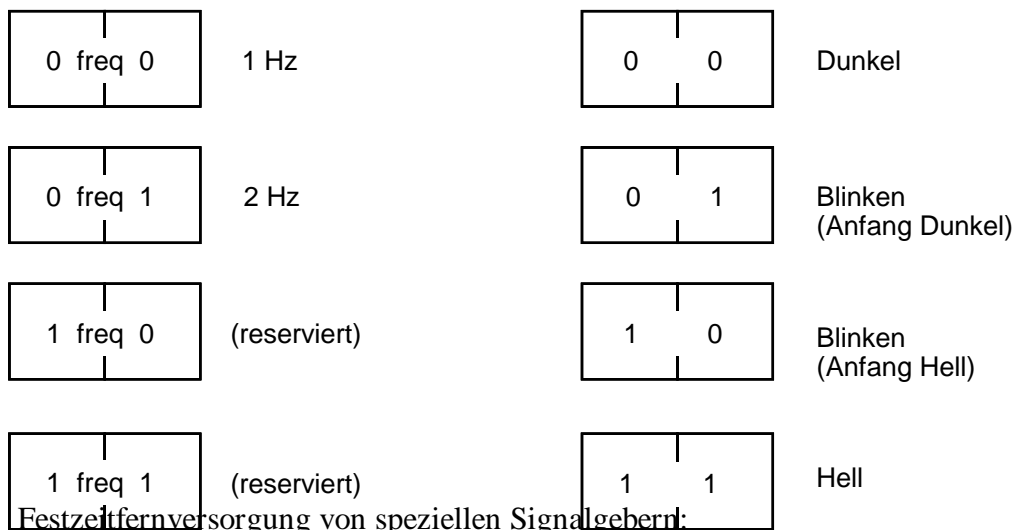
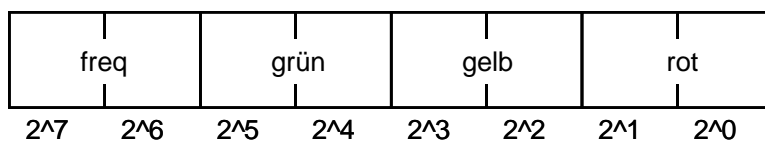
AESignalbild (1:433)

AESignalbild		
METHOD	Name	Beschreibung
150	GetTriggerValue	Liest den Wert, der für die Abtaständerung verwendet wird.
	Eingabeparameter	
		Keine
	Ausgabeparameter	

AESignalbild		
METHOD	Name	Beschreibung
	RetCode	OK : wird zurückgeliefert, wenn das Auftragsselement einen Triggerwert liefern konnte.
	TriggerValue : LONG	Wert des Triggers: aktuelles Signalbild
154	SetSignalgruppe	Setzt die Referenz auf die Signalgruppe.
	Eingabeparameter	
	SignalGruppe : SignalGruppe &	Referenz auf Signalgruppe, Pfad besteht aus: - relativer Knotennummer der Signalgruppe und - Nummer der Signalgruppe.
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragsselement hinzugefügt werden konnte PARAM_INVALID : wenn die Signalgruppe nicht existent ist.

3.5.3.8.1 Signalbildcodierung

Signalbildcodierung Standard: UBYTE:



ÖV-4-Punkt (Badehose):

Wird realisiert mittels 3 Signalgruppen. Versorgungsfarbcode: Rt, Gn oder Gelbblinker

Hüpflicht bzw. „Springlicht“:

Wird realisiert wie ein Blinker. Versorgungsfarbcode: Rt-Gn Wechselblinker

3.5.3.9 Auftragselement DigAusgang (AEDigAusgang)

Dieses Objekt ist für die Online-Visualisierung des Signalplans (z.B. Fußgängertaster) vorgesehen, andere Verwendungszwecke sind optional.

„Digitale Ausgänge“ sind alle Ausgänge, die nicht über Signalschalter arbeiten. Das Auftragselement DigAusgang referenziert immer logische Zustände.

AEDigAusgang ist abgeleitet von AuftragsElement. In der folgenden Tabelle werden deshalb nur die Unterschiede dazu aufgeführt.

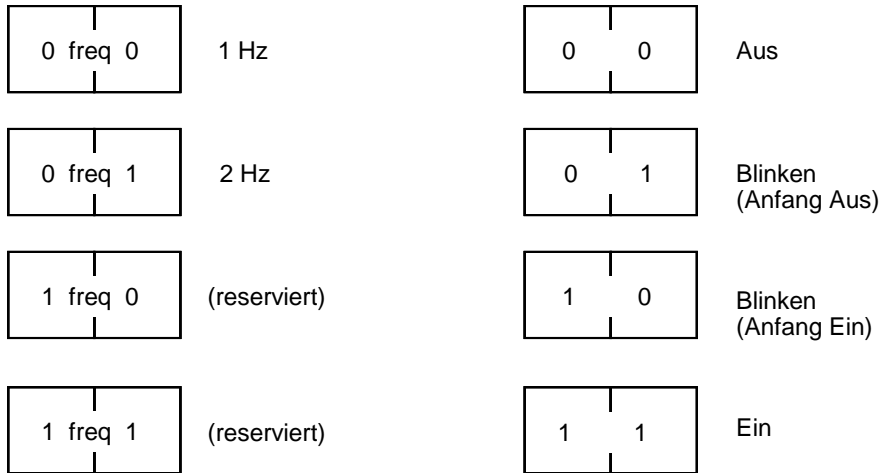
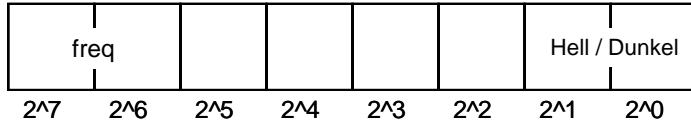
AEDigAusgang (1:439)

AEDigAusgang		
METHOD	Name	Beschreibung
150	GetTriggerValue	Liest den Wert, der für die Abtaständerung verwendet wird
	Eingabeparameter	
		Keine
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement einen Triggerwert liefern konnte.
	TriggerValue : LONG	Wert des Triggers: aktueller Zustand des Ausgangs
154	SetDigAusgang	Setzt die Referenz auf den digitalen Ausgang.
	Eingabeparameter	
	DigAusgang : DigAusgang &	Referenz auf digitalen Ausgang, Pfad besteht aus: - relativer Knotennummer des digitalen Ausgangs und - Nummer des digitalen Ausgangs.
	Ausgabeparameter	
	RetCode	OK : wird zurückgeliefert, wenn das Auftragselement hinzugefügt werden konnte PARAM_INVALID : wenn der digitale Ausgang nicht existent ist.

3.5.3.9.1 Codierung

Codierung Standard: UBYTE:

Die Bits 2^2 bis 2^5 werden nicht benutzt.



3.5.3.10 Kombinationen von Aufträgen und Auftrags-elementen

Vom Gerät unterstützte Kombinationen: x

Nicht sinnvolle Kombinationen: (x)

Diese Kombinationen müssen vom Gerät nicht unterstützt werden und werden daher auch bei Interoperabilitätstests nicht geprüft.

Aufträge	Auftrags-elemente								
	Binär	Aggreg.	Signalbild	APWert	DetExt	AggrExt	APWertVektor	SiptOnline	DigAusg
ZYKL	(x)	x	x	x		x	x	x	x
ABTASTAB	x								
ABTASTÄNDERUNG	(x)	x	x	x		x	x	x	x
VERGLEICH	(x)	x	x	x *)		x	(x)	(x)	x
EXTERN	Keine Auftrags-elemente								
R09	Keine Auftrags-elemente								

Aufträge	Auftragselemente								
	Binär	Aggreg.	Signalbild	APWert	DetExt	AggrExt	APWertVektor	SipOnline	DigAusg
AMLI	Keine Auftragselemente								
DETEXT					x				
MELDUNGSAUFTRAG	Keine Auftragselemente								

*) Vergleich nicht sinnvoll bei APWertBlock

3.5.4 AP-Werte

Anwenderprogrammwerte (AP-Werte) sind innerhalb eines Gerätes durch einen Namen vom Typ der speziellen Domain ANYPATH eindeutig referenziert. ANYPATH ist ein String der Namenskonventionen folgt.

Festlegungen zu den AP-Werten:

- Projektspezifische AP-Werte: Namen von AP-Werten mit systemweiter Bedeutung sind projektspezifisch zu vereinbaren.
- Standard AP-Werte: Festlegungen in Pkt. 3.5.4.5
- AP-Werte der OCIT-I Prozessdaten: Die Namen der AP-Werte die zur Übertragung von Prozessdaten gemäß den OCIT-I Definitionen zu Prozessdaten (Dokument OCIT-I PD-DM) genutzt werden, werden als Strings der Form OCIT-I PD Member.OType gebildet. Zum Beispiel bezeichnet 41.500 die Belegung eines digitalen Eingangs/Detektors. Details siehe OCIT-I Spezifikationen.
- AP-Werte aus VA-Verfahren: Der Verfahrenshersteller muss den OCIT-O Objekttyp der AP-Werte gemäß OCIT-I „Optionale Liste der VA-AP-Werte“ bzw. OCIT-C „intersection_config_data.xsd“ angeben.

Beispiel einer vollständigen Definition eines AP-Wertes:

Beispielwert: AEPWert.SetAP(),Index 1

Attribute	Beispiel Wert	Beschreibung
RefLen	15	Länge der Referenzdefinition in Bytes (ohne RefLen)
Member	1	OCIT-O Member (UShort)
OType	506	OCIT-O Type (UShort)
APWertName	57.101.1	OCIT-O String ANYPATH: entsprechend OCIT-I PD Festlegungen: <OIPM_Member>. <OIPM.Nr>. < Index>; + 2 Bytes Count

Zum Auslesen der Namen der im Lichtsignalsteuergerät verfügbaren AP-Werte stehen folgende Methoden zur Verfügung:

- SOFeldgeraet.InstanceInfo (255 Rückgabewerte) und
- SOFeldgeraet.ExtendedInstanceInfo (65535 Rückgabewerte)

Bei Aufruf von nicht existenten AP-Werten wird ERR_PATH_VAL zurückgegeben.

Hinweis: Bei Lichtsignalsteuergeräten, die sehr viele AP-Werte unterstützen (z.B. VS-Plus Geräte mit mehreren tausend AP-Werten) ist damit zu rechnen, dass die Abfrage der verfügbaren AP-Werte sehr lange dauern kann. Empfehlung: Der Aufruf von SOFeldgeraet.ExtendedInstanceInfo sollte über den Kanal mit niedriger Priorität erfolgen. Reicht auch das nicht aus, so können gezielt einzelne AP-Werte mit komplettem Pfad auf Verfügbarkeit abgefragt werden.

3.5.4.1 APWert (1:505)

Dieser Objekttyp bildet einen Anwenderprogramm Wert (AP-Wert) ab. Anwenderprogrammwerte sind durch Namen eindeutig innerhalb eines Lichtsignalsteuergerätes bestimmt. APWert ist die Basisklasse für AP-Werte. Typen der AP-Werte: USHORT, LONG, BLOB.

APWert (1:505)

APWert		
METHOD	Name	Beschreibung
16	GetWert	Liefert den aktuellen APWert.
	Eingabeparameter	
		Keine
	Ausgabeparameter	
	RetCode	Falls OK gilt folgender Wert.
	Wert : Typ des AP-Werts	Aktueller Anwenderprogramm-Wert. (Prozessvariable)
17	SetWert	Setzt diesen APWert (falls möglich und sinnvoll).
	Eingabeparameter	
	Wert : Typ des AP-Werts	Neuer Anwenderprogramm-Wert.
	Ausgabeparameter	
	RetCode	Falls OK wurde Wert akzeptiert. ACCESS_DENIED falls es nicht vorgesehen ist diesen Wert von außen zu verändern.
18	IsWritable	Liefert Informationen, ob der AP-Wert schreibbar ist.
	Ausgabeparameter	
	RetCode	OK

APWert		
METHOD	Name	Beschreibung
	BOOL	Ist true, falls SetWert möglich ist, false sonst.

3.5.4.1.1 APWertUshort (1:506)

Der Objekttyp Anwenderprogramm-Wert APWertUshort bildet einen AP-Wert vom Typ USHORT ab.

3.5.4.1.2 APWertLong (1:507)

Der Objekttyp **APWertLong** bildet einen Anwenderprogramm Wert vom Typ LONG ab und wird in Auftragsframes mit 4 Byte abgespeichert.

3.5.4.2 APWertRk (1:510)

Der Objekttyp APWertRk bildet relative Knoten bezogene Anwenderprogramm Werte ab. Eine Instanz vom Typ APWertRk wird innerhalb eines Lichtsignalsteuergerätes durch APWertName (String) und RelKnotenNr (UBYTE) eindeutig referenziert.

3.5.4.2.1 APWertRkUshort (1:511)

Der Objekttyp **APWertRkUshort** bildet einen relative Knoten bezogenen Anwenderprogramm Wert vom Typ USHORT ab und wird in Auftragsframes mit 2 Byte abgespeichert.

3.5.4.2.2 APWertRkLong (1:512)

Der Objekttyp **APWertRkLong** bildet einen relative Knoten bezogenen Anwenderprogramm Wert vom Typ LONG ab und wird in Auftragsframes mit 4 Byte abgespeichert.

3.5.4.3 APWertBlock (1:508)

Der Objekttyp APWertBlock bildet einen Anwenderprogramm Wert vom Typ BLOB ab.

Anm: BLOB ist definiert in OCIT-O-Protokoll ab Version 1.0, Kapitel Grunddatentypen:

BLOB	struct{ ULONG sz, BYTE data[] }	Binary large object, in dem die Daten opaque übertragen werden.
------	---------------------------------	---

APWertBlock (1:508)

APWertBlock		
METHOD	Name	Beschreibung
16	GetWert	liefert den aktuellen APWert.

APWertBlock		
METH OD	Name	Beschreibung
	Eingabeparameter	
		(keine)
	Ausgabeparameter	
	RetCode	falls OK gilt folgender Wert.
	Wert : BLOB	Aktueller Anwenderprogramm-Wert.
17	SetWert	Setzt diesen APWert (falls möglich und sinnvoll).
	Eingabeparameter	
	Wert : BLOB	Neuer Anwenderprogramm-Wert.
	Ausgabeparameter	
	RetCode	falls OK wurde Wert akzeptiert. ACCESS_DENIED falls es nicht vorgesehen ist diesen Wert von außen zu verändern.

3.5.4.3.1 APWertRkBlock (1:513)

Der Objekttyp APWertRkBlock bildet einen auf relative Knoten bezogenen Anwenderprogramm Wert vom Typ BLOB.

3.5.4.4 APWertGroup (1:515)

Die Unterstützung von APWertGroup ist in OCIT-O Lstg V2.0 optional.

Mit APWertGroup lassen sich Gruppen von Anwenderprogrammwerten (AP-Werten) bilden. Sie werden genutzt, um komplexe Datenstrukturen oder auch Arrays auf einfache Art und Weise umzusetzen. Das Objekt enthält Methoden, mit denen die in der Gruppe enthaltenen Subgruppen und AP-Werte gelesen werden können.

Um die AP-Werte den Gruppen (APWertGroups) zuzuordnen steht als Strukturierungselement in den Namen der AP-Werte der Punkt „.“ zu Verfügung (analog z.B. zu den Domain-Namen des Internet). Der Name des AP-Werts wird dann gebildet, indem die Namen der enthaltenden APWertGroup durch den Punkt verkettet werden.

Die Gruppennamen ergeben sich automatisch aus den Namen der vorhandenen APWerte und benötigen keine Versorgung.

Für die lichtsignalsteuergerätebezogenen Gruppen gibt es jeweils eine ausgezeichnete Wurzelgruppe mit dem Leerstring als Namen. Diese Wurzelgruppen dienen dann als Ausgangsobjekt für die Abfragen GetElements.

Eine Beschränkung der Schachtelungstiefe der APWertGroups ergibt sich durch die maximale Länge der APWert-Namen (512).

Beispiele:

1. Es gibt die AP-Werte "Geraet.SG1.gruenMin", "Geraet.SG1.gruenMax", "Geraet.SG2.gruenMin", "Geraet.SG2.gruenMax". Die APWertGroup Geraet hat also 2 AP-Wert-Subgruppen Geraet.SG1 und Geraet.SG2, die jeweils 2 AP-Werte enthalten.

2. Beispiel einer vollständigen Definition einer AP-Werte Gruppe:

Beispielwert: AEAPWertVektor.SetListe(), Index 1,5

Attribute	Beispiel Wert	Beschreibung
Prefix	57.101.	OCIT-O String ANYPATH: entsprechend den OCIT-IP Festlegungen zum OIPM-Prefix: <OIPM_Member>. <OIPM.Nr>
Anzahl	2	Anzahl der Elemente in nachfolgender Liste
RefLen	8	Länge der Referentdefinition in Bytes (ohne RefLen)
Member	1	OCIT-O Member (UShort)
OType	506	OCIT-O Type (UShort)
APWertName	1	OCIT-O String ANYPATH: <Index> (4 Byte)
RefLen	8	Länge der Referentdefinition in Bytes (ohne RefLen)
Member	1	OCIT-O Member (UShort)
OType	506	OCIT-O Type (UShort)
APWertName	5	OCIT-O String ANYPATH: <Index> (4 Byte)

APWertGroup (1:515)

APWertGroup		
METHOD	Name	Beschreibung
100	GetElements	Liefert Referenzen auf die APWerte, die in der Gruppe enthalten sind. Dabei werden nicht die Referenzen auf die APWerte geliefert, die in den Subgruppen enthalten sind.
	Ausgabeparameter	
	RetCode	OK

APWertGroup		
METHOD	Name	Beschreibung
	Refs: Path[]	Liste mit Referenzen auf APWerte, die besteht aus Refs.Anzahl Refs[].RefLen Refs[].Member Refs[].OTtype Refs[]. ... Pfadparameter je nach APWert
101	GetSubGroups	Liefert Referenzen auf die Subgruppen der APWertGroup.
	Ausgabeparameter	
	RetCode	OK
	Refs: Path[]	Liste mit Referenzen auf APWertGroups, die besteht aus Refs.Anzahl Refs[].RefLen Refs[].Member Refs[].OTtype Refs[]. ... Pfadparameter je nach APWertGroup

Beispiel: Es gibt eine Gruppe „Det“ mit folgenden Elementen:
Det.Anzahl, Det.Luecke.1, Det.Luecke.2, Det.Belgrad.1, Det.Belgrad.2

Die Methode GetElements liefert: Referenzen auf APWertGroups für Det.Anzahl
Die Methode GetSubgroups liefert: Referenzen APWertGroups für Det.Luecke und Det.Belgrad

3.5.4.4.1 APWertGroupRk (1:516)

Die Unterstützung von APWertGroupRk ist in OCIT-O Lstg V2.0 optional.

Der Objekttyp APWertGroupRk bildet die auf einen relativen Knoten bezogene Gruppen von Anwenderprogrammwerten ab. Für die Rk-bezogenen Gruppen gibt es jeweils eine ausgezeichnete Wurzelgruppe mit dem Leerstring als Namen. Diese Wurzelgruppen dienen dann als Ausgangsobjekt für die Abfragen GetElements.

3.5.4.5 Standardisierte AP-Werte

Häufig verwendete AP-Werte sind in OCIT-O standardisiert.

Werte des Signalplanumlaufs

Umlaufsekunde (TX), laufende Phase (PH) und gewünschte Phase (UE) werden als relative Knoten spezifische AP-Werte (Objektyp: APWertRkUshort) standardisiert.

Umlaufsekunde (TX):

Die Umlaufsekunde des laufenden Signalprogramms wird beginnend mit Sekunde 0 in 100ms-Schritten gezählt. Bei ausgeschaltetem Knoten bleibt der letzte Wert erhalten (Persistenz bei NetzAus nicht nötig). Erstinitialisierung (Reset) mit 0.

Laufende Phase (PH):

0	Phase im Verfahren nicht definiert (z.B. VSPlus)
1..65534	aktuelle Phasennummer

Gewünschte Phase (UE):

0	Es ist kein Phasenübergang aktiv bzw. eine Phase ist im Verfahren nicht definiert (VSPlus)
1.. 65534	Es ist ein Phasenübergang aktiv von Phase PH nach Phase UE.

3.5.4.6 Verfahrensparameter

Dynamische Parameter für ein verkehrstechnisches Verfahren (DPV1)

Die dynamischen Parameter für ein verkehrstechnisches Verfahren, zum Beispiel eine Netzsteuerung, werden als AP-Werte Typ APWertBlock / APWertRkBlock (BLOB) übertragen. Die Inhalte sind anwendungsspezifisch und in OCIT-O nicht standardisiert.

Durch die Namenskonvention DPV1 wird lediglich das im Lichtsignalsteuergerät vorzugsweise angewendete verkehrstechnisches Verfahren als Empfänger der Binärwerte adressiert.

Sollen weitere Parameter dieses Typs verwendet werden, sind ihre Namen projektspezifisch festzulegen.

3.5.5 Detektoren und Signale

3.5.5.1 Digitaler Eingang (DigEingang)

Digitaler Eingang ist das Basisobjekt für digitale Eingänge. Er dient zur Darstellung der Referenzen in der XML-Metabeschreibung, sowie zur Abfrage der binären Zustände und der Namen der digitaler Eingänge. Die Versorgung mit Namen erfolgt proprietär durch den Gerätehersteller.

In den Festlegungen zu OCIT-O referenziert DigEingang auf Detektor-Meldungseingänge und Anforderungssignale. Projektspezifisch kann DigEingang zur Abfrage von frei wählbaren binären, digitalen Signalen verwendet werden.

DigEingang (1:500)

DigEingang		
METHOD	Name	Beschreibung
0	Get	
	Eingabeparameter	
		(keine)
	Ausgabeparameter	
	RetCode	falls OK gilt folgender Wert.
	Name: STRING	Liefert den Namen des dig. Eingangs.
16	GetWert	Liefert den aktuellen Wert.
	Eingabeparameter	
		Keine
	Ausgabeparameter	
	RetCode	Falls OK gilt folgender Wert.
Wert : BOOL=UBYTE	Aktueller Wert	

3.5.5.2 Signalgruppe (SignalGruppe)

Signalgruppe ist das Basisobjekt für Signalgruppen und dient zur Darstellung der Referenzen in der XML-Metabeschreibung, sowie zur Abfrage der Namen der Signalgruppen.

Das Objekt kann mit OCIT-O Methoden nicht geschaltet oder auf binäre Zustände abgefragt werden! Die Versorgung mit Namen erfolgt proprietär durch den Gerätehersteller.

SignalGruppe (1:501)

SignalGruppe		
METHOD	Name	Beschreibung
0	Get	
	Eingabeparameter	
		(keine)
	Ausgabeparameter	
	RetCode	falls OK gilt folgender Wert.
	Name: STRING	Liefert den Namen der Signalgruppe

3.5.5.3 Signalgeber (SignalGeber)

Signalgeber ist das Basisobjekt für Signalgeber und dient zur Darstellung der Referenzen in der XML-Metabeschreibung , sowie zur Abfrage der Namen der Signalgeber.

Da die Namen der Signalgeber von OCIT-I nicht verlangt werden, ist die Methode Get und die Versorgung mit Namen optional.

Das Objekt kann mit OCIT-O Methoden nicht geschaltet oder auf binäre Zustände abgefragt werden! Die Versorgung mit Namen erfolgt proprietär durch den Gerätehersteller.

SignalGeber (1:502)

SignalGeber		
METHOD	Name	Beschreibung
0	Get	
	Eingabeparameter	
		(keine)
	Ausgabeparameter	
	RetCode	falls OK gilt folgender Wert.
	Name: STRING	Liefert den Namen der Signalgeber

3.5.5.4 Signalkammer (SignalKammer)

Signalkammer ist das Basisobjekt für die Signalkammern (Lampen) und dient zur Darstellung der Referenzen in der XML-Metabeschreibung, sowie zur Abfrage der Namen der Signalkammern (Lampen).

Das Objekt kann mit OCIT-O Methoden nicht geschaltet oder auf binäre Zustände abgefragt werden! Die Versorgung mit Namen erfolgt proprietär durch den Gerätehersteller.

Hinweis: Im Standard sind für eine Signalgruppe nur die drei Kammern (0=rot, 1=gelb, 2=grün) definiert. Parallel geschaltete Lampen ändern nicht am Zustand der Signalgruppe. Die Verwaltung / Überwachung der Lampen ist Aufgabe der Signalsicherung ab und steht damit außerhalb der OCIT-O Spezifikation.

SignalKammer (1:503)

SignalKammer		
METHOD	Name	Beschreibung
0	Get	
	Eingabeparameter	
		(keine)
	Ausgabeparameter	
	RetCode	falls OK gilt folgender Wert.
	Name: STRING	Liefert den Namen der Signalkammer (Lampe)

3.5.5.5 Digitaler Ausgang (DigAusgang)

Digitaler Ausgang ist das Basisobjekt für digitale Ausgänge und dient zur Darstellung der Referenzen in der XML-Metabeschreibung, sowie zur Abfrage der Namen der digitalen Ausgänge.

Das Objekt kann mit OCIT-O Methoden nicht geschaltet oder auf binäre Zustände abgefragt werden! Die Versorgung mit Namen erfolgt proprietär durch den Gerätehersteller.

DigAusgang (1:504)

DigAusgang (1:504)		
METHOD	Name	Beschreibung
15	Get	
	Eingabeparameter	
		(keine)
	Ausgabeparameter	
	RetCode	falls OK gilt folgender Wert.
	Name: STRING	Liefert den Namen der Dig. Ausgangs

3.5.6 Archive der Lichtsignalsteuergeräte

Folgende Archive sind in OCIT-Outstations für jedes Lichtsignalsteuergerät fest vorgegeben (sofern benötigt, wie Archive für ÖPNV oder Messwerte):

- Das Betriebszustandsarchiv (0) für die Speicherung des Betriebszustands (Meldungen BzIstvektor). Jede Änderung des IstVektors generiert einen Eintrag des IstVektors in der Betriebszustandsliste. Die Aufträge dazu sind vordefiniert und können nicht geändert werden. Die im Archiv gespeicherten Daten bleiben nach dem Ausschalten der Versorgungsspannung erhalten.
- Das Standard-Meldearchiv (1) enthält Meldungen der Signalsicherung, Störungen und andere Meldungen: OCIT-Meldung + Zusatzmeldung + Meldungsdegree. Die Aufträge dazu sind vordefiniert und können nicht geändert werden. Die im Archiv gespeicherten Daten bleiben nach dem Ausschalten der Versorgungsspannung erhalten.
- Das Syslog-Archiv (2) für Syslog-Meldungen (mit Text) und herstellerspezifischen Meldungen, die persistent gehalten werden. Das Archiv ist bereits im Grundausbau vorhanden. Die Archivgröße wird vom Hersteller an die im Gerät vorhandenen anderen Archive angepasst. Die im Archiv gespeicherten Daten bleiben nach dem Ausschalten der Versorgungsspannung erhalten.
- Ein Archiv Service-Systemzugang (3) für Aufträge die über die Systemzugänge erfolgen.
- Das Versorgungsarchiv (4) für Versorgungsmeldungen.
- Ein Dynamisches Archiv (31) ist für Prozessdaten vorgesehen, deren Beauftragung häufig verändert wird.
- Ein Signalisierungsarchiv (32) für Signalisierungszustände (erfasst bei jedem Zustandswechsel). Mögliche Ergänzungen sind Umlaufsekunde TX, Detektorsignale, Phasen u. a.
- ÖPNV-Archiv (33) für R09-Standard-Telegramme (Erstellungszeitpunkt, Meldepunkt, Linie, Kurs, Route, Priorität, Zuglänge Richtung Hand, Fahrplanabweichung) oder erweiterte R09-Telegramme. Es werden alle für das Lichtsignalsteuergerät relevanten R09-Telegramme im Archiv abgespeichert. Irrelevante Telegramme, die trotzdem empfangen wurden, werden nicht gespeichert.
- Ein Messwertarchiv (34) für aggregierte Detektorwerte wie Fzg/h, Belegung in % und projektspezifische Messwerte.
- Ein Onlinearchiv (35) für Detektorrohwerte (Änderungen des Detektorausgangs) und AP-Werte. Die Abtastintervalle in denen die Änderungen erfasst werden (Auflösung) sind von der Zentrale aus einstellbar. Die höchste einstellbare Auflösung beträgt 10 ms. Wird ein Abtastintervall gewählt, das das Gerät nicht liefern kann, wird eine Fehlermeldung abgesetzt, die auch das vom Gerät unterstützte Intervall beinhaltet.

- Offlinearchiv (36) ist für das OCIT-O Profil 2 vorgesehen.

Die Mindestgrößen der Archive OCIT-konformer Lichtsignalsteuergeräte sind im Dokument Funktionsspiegel (OCIT-O_V1.1_Funktionsspiegel_V1.0) angegeben.

Die Archive 31, 32, 33, 34, 35, 36 sind zur Laufzeit von der Zentrale aus konfigurierbar.

Verhalten bei Stromausfall oder dem Ausschalten der Versorgungsspannung:

- Archive 0, 1, 2, 3, 4: Die im Archiv gespeicherten Daten und die Listenstruktur bleiben erhalten.
- Die Inhalte der nicht persistenten Archive (je nach Einstellung) und die zugehörigen Listen können verloren gehen. Bei Wiederanlauf des Lichtsignalsteuergerätes werden die nicht persistenten Listen zurückgesetzt.

Hinweise:

1. Die festdefinierten Archive 0 bis 4 und 31 bis 36 (siehe Pkt. 3.5.6.3) sollen vorzugsweise und ausschließlich die dafür vorgesehenen Aufträge enthalten.

2. Aufgrund der Belastung der Übertragungsstrecke zwischen Lichtsignalsteuergerät und Zentrale (PD-Server) muss die Übertragung von AP-Werten (ohne Signalisierungsinformationen und Detektorwerte) eingeschränkt werden. Bei typischer Nutzung des Lichtsignalsteuergerätes:

- Profil 1 mit Übertragungsrate von 19200 Baud
- Übertragung von max. 20 SG, max. 32 Detektoren

sollen zusätzlich nicht mehr als 20 AP-Werte sekundlich übertragen werden.

3.5.6.1 Elementbeschreibungen Meldungsarchiv

Das allgemeine Archiv wird pro Meldung als eine Liste von Meldungsteilen übertragen; bei vielen Meldungen fällt nur der an. Eine vollständige Implementierung der hier definierten Fehlermeldungen wird nicht gefordert, da manche Fehlertypen bei manchen LSA nicht auftreten können. Es wird lediglich gefordert, dass die auftretenden Fehler OCIT-Outstations konform codiert werden. Zusätzlich sind auch noch hersteller- oder projektspezifische Meldungsteile oder Meldungen möglich.

Member =1:

(MeldungsDegree I: Information, W: Warnung, F: Fehler, S: Schwerer Fehler)

OType	Kurzname	MeldungsDegree	Beschreibung
60004	Sollbild-Störung	S	ist die Meldung bei Störungen der Sollbildüberwachung. Sehr häufig durch zusätzliche Meldungsteile genauer spezifiziert.
60005	Istbildfehler (schwer)	S	Geräteabschaltung durch unzulässiges Istbild. ohne Rotlampenfehler.
60006	Feindlichkeit	W	wird vom Gerät gemeldet, wenn die Firmware eine Feindlichkeitsverletzung erkennt und korrigiert. Bei Feindlichkeitsverletzungen, die die Signalüberwachung erkennt, wird dieser Meldungsteil als Zusatzmeldungsteil einer Sollbild-Störung gespeichert
60007	Zwischenzeit	W	wird vom Gerät gemeldet, wenn die Firmware eine Zwischenzeitverletzung erkennt und korrigiert. Bei Zwischenzeitverletzungen, die die Signalüberwachung erkennt, wird dieser Meldungsteil als Zusatzmeldungsteil einer Sollbild-Störung gespeichert.
60008	Mindestgrün	W	wird vom Gerät gemeldet, wenn die Firmware eine Mindestgrünzeitverletzung erkennt und korrigiert. Bei Mindestgrünzeitverletzungen, die die Signalüberwachung erkennt, wird dieser Meldungsteil als Zusatzmeldungsteil einer Sollbild-Störung gespeichert.
60009	Mindestrot	W	wird vom Gerät gemeldet, wenn die Firmware eine Mindestrotzeitverletzung erkennt und korrigiert. Bei Mindestrotzeitverletzungen, die die Signalüberwachung erkennt, wird dieser Meldungsteil als Zusatzmeldungsteil einer Sollbild-Störung gespeichert.
60010	Rotlampenfehler	S	Geräteabschaltung durch unzulässiges Istbild aufgrund eines Rotlampenfehlers.
60011	Istbildfehler (sekundär)	W	unzulässiges Istbild (sekundärer Lampenfehler)
60014	Detektorstörung	F	wird eingetragen, wenn ein Detektor ausgefallen ist oder wenn die

OType	Kurzname	MeldungsDegree	Beschreibung
			Plausibilitätskontrolle ausgelöst hat.
60015	Detektor ok	I	wird eingetragen, wenn ein Detektor wieder behoben ist
60022	Versorgung Beginn	I	Eine Versorgungsänderung beginnt. Sehr häufig durch zusätzliche Meldungsteile genauer spezifiziert
60023	VersorgungEnde	I	Eine Versorgungsänderung endet. Sehr häufig durch zusätzliche Meldungsteile genauer spezifiziert
60024	Umlaufkontrolle	F	Wird gemeldet wenn die Umlaufkontrolle ausgelöst wird (gibt es – abhängig vom Steuerverfahren - nicht immer)
60025	Wechsel der Betriebsart	W	Wird gemeldet wenn die Betriebsart gewechselt wird: - Sonderbetrieb - Eigensteuerung (lokale VA-Programmwechsel) - Handstoppbetrieb - lokal fixes Programm - lokal DCF - Zentrale
60037	ÖVempfänger- störung	E	Störung beim Empfang von ÖV-Telegrammen
60038	ÖVempfänger Ok	I	Empfangsstörung ÖV-Telegramme behoben

Bei den Meldungen IstBildfehler, Lampenausfall hat der Parametersatz der Meldung folgenden Aufbau:

RelKnotenNr (UBYTE)	Nummer des relativen Knotens
SigGrpNr (UBYTE)	Nummer der Signalgruppe
SigGeberNr (STRING)	Nummer des Signalgebers
KammerNr (UBYTE)	Nummer der Signalkammer
SigGrpName (STRING)	Name der Signalgruppe
SigGeberName (STRING)	Name des Signalgebers

Bei den Meldungen Mindestgrünzeitunterschreitung und versuchte Mindestgrünzeitunterschreitung hat der Parametersatz der Meldung folgenden Aufbau:

RelKnotenNr (UBYTE)	Nummer des relativen Knotens
SigGrpNr (UBYTE)	Nummer der Signalgruppe
SigGrpName (STRING)	Name der Signalgruppe

Bei den Meldungen (versuchte) Feindlichkeit und (versuchte) Zwischenzeitverletzung hat der Parametersatz der Meldung folgenden Aufbau:

RelKnotenNr (UBYTE)	Nummer des relativen Knotens
SigGrpNrA (UBYTE)	Nummer der einfahrenden Signalgruppe
SigGrpNameA (STRING)	Name der einfahrenden Signalgruppe
SigGrpNrB (UBYTE)	Nummer der räumenden Signalgruppe
SigGrpNameB (STRING)	Name der räumenden Signalgruppe

Bei den Meldungen Detektorstörung und Detektor ok gibt es folgende Parameter (Achtung: Die Detektoren sind in OCIT-Outstations nicht bezogen auf den relativen Knoten):

DetektorNr (UBYTE)	Nummer des Detektors
DetektorName (STRING)	Name des Detektors

Hinweis: Alle Meldungen, die die Versorgung betreffen werden nur ins Versorgungsarchiv eingetragen. Im Standard-Meldearchiv wird jeder Versorgungsvorgang nur mit den Meldungen „Versorgungsbeginn“ und „Versorgungsende“ vermerkt. Handhabung über Include/Exclude-Liste von Standard-Melde- und Versorgungsarchiv.

3.5.6.1.1 Exclude Liste des Standard-Meldearchivs

Folgende Meldungen befinden sich in der Exclude-Liste des Standard-Meldearchivs, d.h. diese Meldungen werden nicht eingetragen:

1	60301	TransactionDefined
1	60315	APWertChangeRequested
1	60316	APWertChangeCommitted
1	60318	TransactionActivationRequest
1	60319	SupplyVersionChanged

Alle anderen Meldungen befinden sich somit in der Include-Liste und werden eingetragen.

3.5.6.1.2 Include Liste des Versorgungsarchivs

Folgende Meldungen befinden sich in der Include-Liste des Versorgungsarchivs, d.h. diese Meldungen werden eingetragen:

1	60022	VersorgungsBeginn
1	60023	VersorgungsEnde
1	60301	TransactionDefined
1	60315	APWertChangeRequested
1	60316	APWertChangeCommitted
1	60318	TransactionActivationRequest
1	60319	SupplyVersionChanged

Alle anderen Meldungen befinden sich somit in der Exclude-Liste und werden nicht eingetragen.

3.5.6.2 Elementbeschreibungen Betriebszustandsarchiv

Das Betriebszustandsarchiv wird pro Meldung als eine Liste von Meldungsteilen übertragen, bei denen jeweils die Vorgangskennung im Meldungsteil und der „neue“ Wert als Datensatz übertragen wird.

Es müssen in der BZ-Meldung auch die BZ-Meldungsteile übertragen werden, die sich nicht ändern! Dadurch wird sichergestellt, dass auch bei verloren gegangenen Meldungen immer der Gesamtzustand des Systems dargestellt werden kann.

Theoretisch lassen sich die Meldungsteile auch in anderen Archiven getrennt eintragen. Dieses Verfahren ist nicht verboten, das Betriebszustandsarchiv ist jedoch unbedingt zu implementieren.

Pro relativem Knoten gibt es einen Meldungsteil der den IstVektor (s. 3.4.19) enthält. Diese Meldung kann um herstellerspezifische Meldungsteile erweitert werden.

Meldungsteil für den knotenbezogenen Betriebszustand

Name	Beschreibung
Knotennummer	Verweis auf relativen Knoten, der diesen Eintrag veranlasst. (Alle folgenden Nummern gelten für diesen Knoten)
Sammelstoerung : ui1	0= keine Störung 1= Störung ohne Abschaltung 2= Störung mit Abschaltung 3= Störung mit Teilabschaltung 4= Interne Störung ohne Abschaltung
IBetriebsart	
Vorgangskennung	Vorgangskennung der Betriebsartwahl welche zum Erreichen der folgenden Betriebsart führte.
Betriebsart : UBYTE	Sonderbetrieb Eigensteuerung (lokale VA-Programmwechsel) Handstoppbetrieb lokal fixes Programm lokal Zeitsteuerung Zentrale (Gerät berücksichtigt Zentralenschaltwunsch)
ISignalProgramm	
Vorgangskennung	Vorgangskennung der Signalprogrammwahl welche zum Erreichen der folgenden SigProgNr führte.
SigProgNr	Zu dieser Zeit eingestellte Signalprogrammnummer
IKnotenEinAus	
Vorgangskennung	Kennung des Vorgangs, welcher zu folgendem KZustand führte.
KZustand	Ein/Auszustand des gesamten relativen Knotens
ITeilknoten[]	
Vorgangskennung	
TKZustand	Zu dieser Zeit eingestellter Teilknotenzustand
ISondereingriff	
Vorgangskennung	
SondereingriffNr	eingestellter Sondereingriff
Modifikations[0..15]	

Name	Beschreibung
	Modifikationen des Signalprogramms. Hier kann jede von IModEinAus abgeleitete Klasse stehen, dies sind z. Zt. IVAEinAus, IOepnvEinAus, IProjEinAus. Übertragung als Array mit variablen Typen.
Vorgangskennung	Kennung des Vorgangs, welcher zu diesem Schaltwunsch führte
Zustand	Zustand der Modifikation

3.5.6.3 Eigenschaften der Listen

Listennummer	0 Betriebszustand	1 Standard-Meldearchiv	2 Syslog	3 Service	4 Versorgung	31 Dynamisches Archiv	32 Signali-sierung	33 ÖPNV	34 Messwert	35 Online	36 Offline (für Profil 2)
Anlegen von Aufträgen möglich?	Nein	Nein	Ja	Ja	Nein	Ja	Ja	Ja	Ja	Ja	Ja
Start/Stop / Reset der Liste möglich?	Nein	Nein	Ja	Ja	Nein	Ja	Ja	Ja	Ja	Ja	Ja
Suspend / Unsuspend der Liste möglich?	Nein	Nein	Ja	Ja	Nein	Ja	Ja	Ja	Ja	Ja	Ja
Vorbelegte Persistenz (keine, Auftrag, Aufträge & Buffer)	Aufträge & Buffer	Aufträge & Buffer	Aufträge & Buffer	Aufträge & Buffer	Aufträge & Buffer	Keine	Keine	Keine	Keine	Keine	Keine
Auswahl der Persistenz möglich	Nein	Nein	Nein	Nein	Nein	hersteller-abhängig	hersteller-abhängig	hersteller-abhängig	hersteller-abhängig	hersteller-abhängig	Ja
Vorbelegter Zustand der Liste (Start, Stop, Suspend)	Start	Start	Stop	Stop	Start	Stop	Stop	Stop	Stop	Stop	Stop
OverwriteOn-Full aktiv?	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Größe des Buffers änderbar?	Nein	Nein	Nein	Nein	Nein	Ja	Ja	Ja	Ja	Ja	Ja
Vordefinierte Aufträge (nach Reset)	Knoten-betriebs-zustand 0:Meldungs-auftrag I	0:Meldungs-auftrag I 1:Meldungs-auftrag W 2:Meldungs-auftrag E 3:Meldungs-auftrag F	0:Meldungs-auftrag I	0:Meldungs-auftrag I	0:Meldungs-auftrag I	0:Meldungs-auftrag I	0:Meldungs-auftrag I	0:Meldungs-auftrag I	0:Meldungs-auftrag I	0:Meldungs-auftrag I	0:Meldungs-auftrag I
											Achtung: Zeitsprung wird nicht eingetragen!

Hinweise:

1. Dynamisches Archiv (Liste 31): Bei jeder Änderung des Auftrags ist damit zu rechnen, dass ein ResetListe generiert wird, was zu einer kurzen Datenunterbrechung führt.
Empfehlung: AddAuftrag bei laufender Liste unterstützen. Falls die Aufträge häufig geändert werden, können die max. 254 Auftragsnummern ausgehen. Dann ist ein ResetListe notwendig.

2. Offlinearchiv(Liste 36): Die Verwendung der Include / Exclude-Methoden ist notwendig. Siehe auch Dokument Profil 2. Die Methoden Include / Exclude sollten vorzugsweise wäh-

rend der Konfiguration aufgerufen werden. Näheres dazu siehe die Definition im Dokument Basis 4.2.6.2 Meldungsauftrag. Vordefinierte Aufträge können nicht verändert werden.

3. Listen mit AP-Werten welche einer Verkehrslogikanwendung zugeordnet sind, müssen nach einer Neuversorgung der Verkehrslogik manuell entsprechend angepasst werden, weil möglicherweise neue Werte dazukommen und **nicht mehr vorhandene** AP-Werte entfernt werden sollen. Der Wert nicht mehr vorhandener AP-Werte wird im Lichtsignalsteuergerät automatisch auf NULLVALUE gesetzt, beim BLOB wird die Länge auf Null gesetzt.

Glossar

Der Inhalt dieses Kapitels führt fachtechnische Begriffe auf, die auf den Kontext dieses Dokuments beziehen. Begriffe, die in allen OCIT-Dokumenten vorkommen können finden sich im Dokument OCIT-O System.

AP-Werte	Überbegriff in OCIT-O für ausgewählte interne Variable des Lichtsignalsteuergerätes, die von Anwenderprogrammen dynamisch gebildet werden oder (wenn setzbar) zur Steuerungen von Anwendungen durch übergeordnete zentrale Applikationen dynamisch verändert werden können.
Archive	In Archiven werden ausgewählte Daten des Lichtsignalsteuergerätes, die zur Dokumentation von Betriebszuständen bzw. zur Speicherung von dynamischen Werten dienen, gesammelt. Das Speicherformat (Bereitstellungsformat) kann vom Format der einzelnen Daten abweichen, um damit eine Datenkomprimierung zu erreichen.
Befehlsquellen	Befehlsquellen sind unterschiedliche Verursacher der Befehle für die Wahl des Signalprogramms oder der Betriebsart.
Betriebsart	Eine Bezeichnung für bestimmte Arten der Steuerung (z.B. lokale oder zentrale Steuerung eines Lichtsignalsteuergerätes).
Betriebszustand	Eine Bezeichnung für einen Zustand, wie z.B. Ein, Aus, Störung.
BTPPL	Basis Transport Paket Protokoll Layer der OCIT-O Schnittstelle
DTD	Document Type Definition Ein Satz an Regeln, der benutzt wird um Dokumente eines bestimmten Typs zu repräsentieren. DTD ist Bestandteil der XML-Spezifikation.
Dynamische Werte	Überbegriff für ausgewählte interne Variable des Lichtsignalsteuergerätes, welche üblicherweise durch Netzsteuerverfahren beeinflusst werden.
Ein/Ausschaltbilder	Eine Folge von Signalisierungszuständen über die ein Gerät von Aus nach Ein in das gewünschte Signalprogramm oder von Ein nach Aus wechselt.
Event	Bestimmte Ereignisse im Lichtsignalsteuergerät lösen eine Nachricht an die Zentrale aus. Diese Nachricht wird als Event bezeichnet. Events werden z. B. ausgelöst, wenn Archive voll sind oder Meldungen von der Zentrale angefordert werden sollen.
Fehlermeldung	Im Gegensatz zu Störungen (Störungsmeldung) sind Fehler nicht durch einen technischen Defekt bedingt, sondern Fehler in der Versorgung (z.B. der Zwischenzeit) oder Bedienung (z.B. nicht ausführbarer Befehl) des Lichtsignalsteuergerätes.
herstellerspezifisch	Der jeweilige Hersteller legt die genaue Systematik oder Funktionsweise fest. In der Regel sind hier keine projektspezifischen Definitionen möglich oder sinnvoll, da sie die Durchgängigkeit und Robustheit der herstellereigenen Lösung gefährden würden.
IP	Internet Protocol (Version 4, soweit nicht anders vermerkt)
ISO / OSI	ISO/OSI-Basis-Referenzmodell (DIN-ISO 7498 v.1982, X.200 v. 1994) ISO: International Organization for Standardization OSI: Open Systems Interconnection

Knoten	Auch: Kreuzung, Knotenpunkt. Sammelbegriff für die unterschiedlichsten Formen von Straßenkreuzungen, d.h. auch Kreisverkehre. Ein Lichtsignalsteuergerät kann je nach Auslegung mehrere Knotenpunkte steuern. Umgekehrt ist es möglich, dass mehrere Lichtsignalsteuergeräte einen Knotenpunkt steuern. In OCIT-Outstations ist festgelegt, dass jeder Knoten auch Teilknoten enthalten kann.
Lstg	Lichtsignalsteuergerät
Meldungen	Meldungen bezeichnen Ereignisse und nennen Verursacher, Zeitpunkt des Auftretens etc. Meldungen werden in Archiven gespeichert (Standard-Meldungsarchiv). Die Zentrale erhält die Meldungen nicht direkt, sondern nur eine Nachricht, dass Meldungen vorliegen (Event), worauf sie die Meldungen vom Lichtsignalsteuergerät anfordert und erhält.
Messwerte	Messwerte sind Messergebnisse der Sensorik und andere vom Gerät erfasste Daten, die als Originalwert oder vorverarbeitet eine Aussage über das Verkehrsgeschehen liefern.
OCIT-Outstations	<ul style="list-style-type: none"> • Komponenten mit OCIT-Schnittstellen auf der Feldebene, die Dienste auf der Feldebene eines Systems der Straßenverkehrstechnik unter Verwendung von OCIT-Schnittstellen bereitstellen.
ÖPNV-Archiv	In das ÖPNV-Archiv werden die ÖPNV-Telegramme, ergänzt mit Werten aus dem Steuergerät, archiviert. Das ÖPNV-Archiv ist ein Spezialfall eines Archivs.
ÖPNV-Telegramm	Auch: ÖV-Telegramm, R09-Telegramm. Das Standard-Telegramm nach R09-xx besteht aus folgenden Datensätzen: Datum, Uhrzeit, Melde-, Linien-, Kurs- und Routennummer, Priorität, Zuglänge, Richtung, Fahrplanabweichung. Optional kann in OCIT das um einige Datensätze erweiterte ÖPNV-Telegramm verwendet werden.
Phasen	Eine Phase ist ein Teil eines Signalzeitenplanes, in dem ein bestimmter Signalisierungszustand unverändert bleibt (während des Beginns einer Phase können noch Phasenübergänge ablaufen). In der Geräteversorgung sind Phasen eine Zuordnung von Schaltzuständen zu Signalgruppen. Den Phasenübergängen werden Schaltzeiten zugeordnet.
projektspezifisch	Die jeweilige Spezifikation lässt in der Regel innerhalb der durch die vorhandene Technik festgelegten Grenzen projektspezifische Systematik oder Funktionen zu.
Relative Knoten	Das Adressierungsschema von OCIT-Outstations sieht vor, dass mit einem Gerät mehrere logisch voneinander unabhängige Knotenpunkte (relative Knoten) realisiert werden können. Nicht alle Hersteller können derartige (aufwändige) Geräte anbieten.
Returncode	Wird von der Zentrale ein Leistungsmerkmal aufgerufen, das im Lichtsignalsteuergerät nicht verfügbar ist, wird ein Returncode erzeugt und übertragen, den die Zentrale auswerten kann.
SHA-1	Secure Hash Algorithm
Signalgruppe	Eine Signalgruppe umfasst all jene Lichtsignale an einem Knotenpunkt, die zu jedem Zeitpunkt in ihrem Signalisierungszustand übereinstimmen.

Signalgruppenversorgung	Signalgruppenversorgung ist ein Teil der Versorgungsdaten. Es handelt sich um die Datenversorgung der Signalgruppentypen. , Farbkombinationen, Ein/Ausschaltbilder u. a. sicherheitsrelevante Daten wie die Zwischenzeiten können im allgemeinen von der Zentrale aus während des normalen Betriebes nicht verändert werden.
Signalisierungszustand	Auch: Signalisierung, Signalzustand, Signalbild. Die an den Signalgebern eines Knotens geschalteten Lichtsignale, die einen bestimmten Zustand an den Signalgruppen ergeben, z.B. Grün, Gelb, Rot, Dunkel, Blinken usw.
Signalplan	Er enthält die Dauer von Signalzeiten und die Zuordnung zu bestimmten Signalgruppen (Signalisierungszustände). Dazu kommen Daten für Synchronisierung und Signalprogrammwechsel. Signalpläne sind ein Teil der Versorgungsdaten für Festzeit- und/oder verkehrsabhängige Steuerverfahren. Sondersignalpläne wie z.B. Feuerwehrpläne sind ebenfalls Signalpläne.
Signalprogramme	Signalprogramme sind Anweisungen für den Steuerungsablauf. Sie bestimmen die zeitliche Folge der Signalisierungszustände auf der Grundlage von Signalplänen und/oder den Logiktyp (Festzeit, Phasen, Verkehrsabhängigkeit). Jedem Signalprogramm sind Ein/ Ausschaltbilder zugeordnet. Der Betriebszustand „Aus“ ist kein Signalprogramm.
Signalzeitenplan	Der Signalzeitenplan ist die graphische Darstellung eines Signalprogramms im Zeitmaßstab.
SiSi	Signalsicherung
Sondereingriff	Auswahl eines nur temporär gültigen Signalprogramms, z.B. eines Feuerwehrplanes. Nach Ende des Sondereingriffes kehrt das Gerät in den ursprünglichen Zustand / Signalprogramm zurück.
Störungsmeldungen	Störungsmeldungen melden das Auftreten einer durch einen technischen Defekt verursachten Störung einer Systemkomponente. Sie beinhalten den Verursacher mit möglichst genauer Lokalisierung des Störungsorts und die Art der Störung (Unterscheidung: Fehlermeldungen).
Synchronisation	Die Synchronisierung in grünen Wellen basiert auf gleichlaufenden Uhren. Das dazu notwendige Rückrechnungsverfahren ist projektspezifisch festzulegen, da das Rückrechnungsverfahren im System (Bestand + OCIT) gleich sein muss.
TCP	Transmission Control Protocol Eines der Internetprotokolle. Verbindungsorientiertes Transportprotokoll in Schicht 4 des ISO/OSI-Referenzmodells.
Teilknoten	Teilknoten sind zu einzelnen Signalisierungsbereichen zusammengefasste Signalgruppen eines Gesamtknotens, die zueinander nicht feindlich sind. Alle Teilknoten arbeiten zu einer bestimmten Zeit mit dem selben Signalprogramm. Teilknoten können von der Zentrale ein- und ausgeschaltet werden.
UDP	User Datagram Protocol Eines der Internetprotokolle. Verbindungsloses Protokoll in Schicht 4 des ISO/OSI-Referenzmodells.
Verkehrstechnische Verfahren (auch verkehrsabhängige Logik, VA-Logik, VA, VA-Verfahren)	Software im Lichtsignalsteuergerät, die auf der Basis vorgegebener Algorithmen und Verkehrsmesswerten die Signalisierung entsprechend der aktuellen Verkehrssituation modifiziert. Die Algorithmen der Logik sind durch Parameter veränderbar (ein Teil der Versorgungsdaten). Berechnete Ergebnisse (Variable) können in OCIT-Outstations als AP-Werte gelesen oder gesetzt werden.

Versorgungsdaten	Versorgungsdaten sind alle statischen und quasistatischen Daten, die das einem Teilsystem eines Lichtsignalsteuerungssystems zugrunde liegende Datenmodell für die Erfüllung seiner Funktionalität benötigt. Beispiele für Versorgungsdaten eines Teilsystems des Anwendungsbereichs Lichtsignalsteuerung sind: Signalgruppen, Signalfolgen, Minimale Freigabezeiten, Maximale Freigabezeiten, Festzeitsignalprogramme, Parameter für verkehrsabhängige Signalprogramme. Versorgungsdaten entstehen vorrangig am Verkehrsingenieurs-Arbeitsplatz und werden in den Teilsystemen eines Lichtsignalsteuerungssystems für die Umsetzung der planerischen Vorgaben der Lichtsignalsteuerung benötigt. Der OCIT-Prozess definiert ein Datenmodell für Versorgungsdaten, unterschieden nach Daten zur Anwenderversorgung und nach Daten zur Herstellerversorgung, das im Dokument OCIT-Instations DM VD dokumentiert ist.
VIAP	Verkehrsingenieurarbeitsplatz Werkzeug zum Planen, Simulieren, Versorgen und Testen der verkehrstechnischen Datenversorgung von Lichtsignalanlagen.
Visualisierungsdaten	Daten, die zur Anzeige von sekundlichen Abläufen an der LSA auf dem Bildschirm der Zentrale dienen. Diese Daten können auch für Analysezwecke benutzt werden.
XML	Extensible Markup Language Metasprache für das Definieren von Dokumenttypen. XML liefert die Regeln, die beim Definieren von Dokumenttypen angewendet werden.
XSD	XML Schema Definition Eine komplexe Schemasprache zur Beschreibung eines XML-Typs. Im Gegensatz zu DTD kann bei Verwendung von XSD zwischen dem Namen des XML-Typs und dem in der Instanz verwendeten XML-Tagnamen unterschieden werden.
Zentrale	Das Wort Zentrale wird in den OCIT-O Dokumenten als Kurzform für eine Lichtsignalsteuerungszentrale an die Lichtsignalsteuergeräte angeschlossen sind, verwendet. Die Lichtsignalsteuerungszentrale kann ein Teil einer aus mehreren Komponenten bestehende Einrichtung zur Steuerung und Überwachung des Straßenverkehrs sein. Die Komponenten dieser zentralen Ebene können sich an verschiedenen Orten befinden (verteiltes System).
Zentrale Ebene	Eine aus einer oder mehreren Komponenten bestehende Einrichtung zur Steuerung und Überwachung des Straßenverkehrs. Die Komponenten der zentralen Ebene können sich an verschiedenen Orten befinden (verteiltes System). Nach dem Verständnis des OCIT-Prozesses umfasst die zentrale Ebene mindestens eine Lichtsignalsteuerungszentrale und die daran angeschlossenen Lichtsignalanlagen mit ihren Lichtsignalsteuergeräten. Erweiterungen sind die Teilsysteme wie Verkehrsingenieurs-Arbeitsplatz, Versorgungsdatenserver, System zur Qualitätssicherung, Adaptive Netzsteuerung und ggf. weitere.
Zentraler und lokaler Systemzugang	OCIT-Outstations Schnittstelle der zentralen Ebene oder am Lichtsignalsteuergerät, an der Werkzeuge für Versorgung oder Service angeschlossen werden können.

OCIT-O_Lstg_V2.0_A04

Copyright © 2012 ODG
