
Европейская экономическая комиссия

Комитет по внутреннему транспорту

Рабочая группа по автомобильному транспорту

**Группа экспертов по Европейскому соглашению,
касающемуся работы экипажей транспортных
средств, производящих международные
автомобильные перевозки (ЕСТР)**

Пятнадцатая сессия

Женева, 12 июня 2017 года

Данный документ, представленный Европейской Комиссией, содержит добавление 1 к приложению IC к регламенту (ЕС) 2016/799.

RU

ПРИЛОЖЕНИЕ 1. СЛОВАРЬ ДАННЫХ

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	10
1.1. Метод определения типов данных	10
1.2. Ссылки	10
2. ОПРЕДЕЛЕНИЯ ТИПОВ ДАННЫХ	12
2.1. ActivityChangeInfo	12
2.2. Address	13
2.3. AESKey	13
2.4. AES128Key	13
2.5. AES192Key	13
2.6. AES256Key	14
2.7. BCDString	14
2.8. CalibrationPurpose	14
2.9. CardActivityDailyRecord	15
2.10. CardActivityLengthRange	15
2.11. CardApprovalNumber	15
2.12. CardCertificate	15
2.13. CardChipIdentification	16
2.14. CardConsecutiveIndex	16
2.15. CardControlActivityDataRecord	16
2.16. CardCurrentUse	16
2.17. CardDriverActivity	17
2.18. CardDrivingLicenceInformation	17
2.19. CardEventData	17
2.20. CardEventRecord	18
2.21. CardFaultData	18
2.22. CardFaultRecord	18
2.23. CardIccIdentification	19
2.24. CardIdentification	19
2.25. CardMACertificate	19
2.26. CardNumber	20
2.27. CardPlaceDailyWorkPeriod	20
2.28. CardPrivateKey	20
2.29. CardPublicKey	20
2.30. CardRenewalIndex	21
2.31. CardReplacementIndex	21
2.32. CardSignCertificate	21

2.33.	CardSlotNumber	21
2.34.	CardSlotsStatus	21
2.35.	CardSlotsStatusRecordArray	21
2.36.	CardStructureVersion	22
2.37.	CardVehicleRecord	22
2.38.	CardVehiclesUsed	23
2.39.	CardVehicleUnitRecord	23
2.40.	CardVehicleUnitsUsed	24
2.41.	Certificate	24
2.42.	CertificateContent	25
2.43.	CertificateHolderAuthorisation	25
2.44.	CertificateRequestID	26
2.45.	CertificationAuthorityKID	26
2.46.	CompanyActivityData	27
2.47.	CompanyActivityType	27
2.48.	CompanyCardApplicationIdentification	27
2.49.	CompanyCardHolderIdentification	28
2.50.	ControlCardApplicationIdentification	28
2.51.	ControlCardControlActivityData	28
2.52.	ControlCardHolderIdentification	29
2.53.	ControlType	29
2.54.	CurrentDateTime	30
2.55.	CurrentDateTimeRecordArray	30
2.56.	DailyPresenceCounter	30
2.57.	Datef	31
2.58.	DateOfDayDownloaded	31
2.59.	DateOfDayDownloadedRecordArray	31
2.60.	Distance	31
2.61.	DriverCardApplicationIdentification	32
2.62.	DriverCardHolderIdentification	33
2.63.	DSRCSecurityData	33
2.64.	EGFCertificate	33
2.65.	EmbedderIcAssemblerId	33
2.66.	EntryTypeDailyWorkPeriod	34
2.67.	EquipmentType	34
2.68.	EuropeanPublicKey	35
2.69.	EventFaultRecordPurpose	35

2.70.	EventFaultType	36
2.71.	ExtendedSealIdentifier	37
2.72.	ExtendedSerialNumber	38
2.73.	FullCardNumber	38
2.74.	FullCardNumberAndGeneration	39
2.75.	Generation	39
2.76.	GeoCoordinates	39
2.77.	GNSSAccuracy	39
2.78.	GNSSContinuousDriving	40
2.79.	GNSSContinuousDrivingRecord	40
2.80.	GNSSPlaceRecord	41
2.81.	HighResOdometer	41
2.82.	HighResTripDistance	41
2.83.	HolderName	41
2.84.	InternalGNSSReceiver	42
2.85.	K-ConstantOfRecordingEquipment	42
2.86.	KeyIdentifier	42
2.87.	KMWCKey	42
2.88.	Language	42
2.89.	LastCardDownload	42
2.90.	LinkCertificate	43
2.91.	L-TyreCircumference	43
2.92.	MAC	43
2.93.	ManualInputFlag	43
2.94.	ManufacturerCode	43
2.95.	ManufacturerSpecificEventFaultData	43
2.96.	MemberStateCertificate	44
2.97.	MemberStateCertificateRecordArray	44
2.98.	MemberStatePublicKey	44
2.99.	Name	44
2.100.	NationAlpha	45
2.101.	NationNumeric	45
2.102.	NoOfCalibrationRecords	45
2.103.	NoOfCalibrationsSinceDownload	45
2.104.	NoOfCardPlaceRecords	45
2.105.	NoOfCardVehicleRecords	45
2.106.	NoOfCardVehicleUnitRecords	46

2.107.	NoOfCompanyActivityRecords _____	46
2.108.	NoOfControlActivityRecords _____	46
2.109.	NoOfEventsPerType _____	46
2.110.	NoOfFaultsPerType _____	46
2.111.	NoOfGNSSCDRecords _____	46
2.112.	NoOfSpecificConditionRecords _____	46
2.113.	OdometerShort _____	46
2.114.	OdometerValueMidnight _____	47
2.115.	OdometerValueMidnightRecordArray _____	47
2.116.	OverspeedNumber _____	47
2.117.	PlaceRecord _____	47
2.118.	PreviousVehicleInfo _____	48
2.119.	PublicKey _____	48
2.120.	RecordType _____	49
2.121.	RegionAlpha _____	49
2.122.	RegionNumeric _____	50
2.123.	RemoteCommunicationModuleSerialNumber _____	51
2.124.	RSAPublicModulus _____	51
2.125.	RSAPrivateExponent _____	51
2.126.	RSAPublicExponent _____	51
2.127.	RtmData _____	51
2.128.	SealDataCard _____	51
2.129.	SealDataVu _____	51
2.130.	SealRecord _____	52
2.131.	SensorApprovalNumber _____	52
2.132.	SensorExternalGNSSApprovalNumber _____	52
2.133.	SensorExternalGNSSCoupledRecord _____	53
2.134.	SensorExternalGNSSIdentification _____	54
2.135.	SensorExternalGNSSInstallation _____	54
2.136.	SensorExternalGNSSOSIdentifier _____	54
2.137.	SensorExternalGNSSSCIIdentifier _____	55
2.138.	SensorGNSSCouplingDate _____	55
2.139.	SensorGNSSSerialNumber _____	55
2.140.	SensorIdentification _____	55
2.141.	SensorInstallation _____	56
2.142.	SensorInstallationSecData _____	56
2.143.	SensorOSIdentifier _____	56

2.144.	SensorPaired	57
2.145.	SensorPairedRecord	57
2.146.	SensorPairingDate	57
2.147.	SensorSCIIdentifier	57
2.148.	SensorSerialNumber	58
2.149.	Signature	58
2.150.	SignatureRecordArray	58
2.151.	SimilarEventsNumber	58
2.152.	SpecificConditionRecord	58
2.153.	SpecificConditions	59
2.154.	SpecificConditionType	59
2.155.	Speed	59
2.156.	SpeedAuthorised	59
2.157.	SpeedAverage	60
2.158.	SpeedMax	60
2.159.	TachographPayload	60
2.160.	TachographPayloadEncrypted	60
2.161.	TDesSessionKey	60
2.162.	TimeReal	60
2.163.	TyreSize	60
2.164.	VehicleIdentificationNumber	61
2.165.	VehicleIdentificationNumberRecordArray	61
2.166.	VehicleRegistrationIdentification	61
2.167.	VehicleRegistrationNumber	61
2.168.	VehicleRegistrationNumberRecordArray	61
2.169.	VuAbility	62
2.170.	VuActivityDailyData	62
2.171.	VuActivityDailyRecordArray	62
2.172.	VuApprovalNumber	63
2.173.	VuCalibrationData	63
2.174.	VuCalibrationRecord	64
2.175.	VuCalibrationRecordArray	65
2.176.	VuCardIWData	65
2.177.	VuCardIWRecord	66
2.178.	VuCardIWRecordArray	67
2.179.	VuCardRecord	67
2.180.	VuCardRecordArray	67

2.181.	VuCertificate	68
2.182.	VuCertificateRecordArray	68
2.183.	VuCompanyLocksData	68
2.184.	VuCompanyLocksRecord	68
2.185.	VuCompanyLocksRecordArray	69
2.186.	VuControlActivityData	69
2.187.	VuControlActivityRecord	70
2.188.	VuControlActivityRecordArray	70
2.189.	VuDataBlockCounter	71
2.190.	VuDetailedSpeedBlock	71
2.191.	VuDetailedSpeedBlockRecordArray	71
2.192.	VuDetailedSpeedData	71
2.193.	VuDownloadablePeriod	72
2.194.	VuDownloadablePeriodRecordArray	72
2.195.	VuDownloadActivityData	72
2.196.	VuDownloadActivityDataRecordArray	73
2.197.	VuEventData	73
2.198.	VuEventRecord	74
2.199.	VuEventRecordArray	75
2.200.	VuFaultData	75
2.201.	VuFaultRecord	76
2.202.	VuFaultRecordArray	77
2.203.	VuGNSSCDRecord	77
2.204.	VuGNSSCDRecordArray	77
2.205.	VuIdentification	78
2.206.	VuIdentificationRecordArray	79
2.207.	VuITSConsentRecord	79
2.208.	VuITSConsentRecordArray	79
2.209.	VuManufacturerAddress	80
2.210.	VuManufacturerName	80
2.211.	VuManufacturingDate	80
2.212.	VuOverSpeedingControlData	80
2.213.	VuOverSpeedingControlDataRecordArray	80
2.214.	VuOverSpeedingEventData	81
2.215.	VuOverSpeedingEventRecord	81
2.216.	VuOverSpeedingEventRecordArray	82
2.217.	VuPartNumber	82

2.218.	VuPlaceDailyWorkPeriodData	83
2.219.	VuPlaceDailyWorkPeriodRecord	83
2.220.	VuPlaceDailyWorkPeriodRecordArray	83
2.221.	VuPrivateKey	84
2.222.	VuPublicKey	84
2.223.	VuSerialNumber	84
2.224.	VuSoftInstallationDate	84
2.225.	VuSoftwareIdentification	84
2.226.	VuSoftwareVersion	84
2.227.	VuSpecificConditionData	85
2.228.	VuSpecificConditionRecordArray	85
2.229.	VuTimeAdjustmentData	85
2.230.	VuTimeAdjustmentGNSSRecord	85
2.231.	VuTimeAdjustmentGNSSRecordArray	86
2.232.	VuTimeAdjustmentRecord	86
2.233.	VuTimeAdjustmentRecordArray	87
2.234.	WorkshopCardApplicationIdentification	88
2.235.	WorkshopCardCalibrationData	89
2.236.	WorkshopCardCalibrationRecord	89
2.237.	WorkshopCardHolderIdentification	90
2.238.	WorkshopCardPIN	90
2.239.	W-VehicleCharacteristicConstant	91
2.240.	VuPowerSupplyInterruptionRecord	91
2.241.	VuPowerSupplyInterruptionRecordArray	91
2.242.	VuSensorExternalGNSSCoupledRecordArray	92
2.243.	VuSensorPairedRecordArray	92
3.	ОПРЕДЕЛЕНИЯ ДИАПАЗОНОВ ЗНАЧЕНИЙ И РАЗМЕРОВ	93
4.	НАБОРЫ СИМВОЛОВ	93
5.	КОДИРОВКА	93
6.	ИДЕНТИФИКАТОРЫ ОБЪЕКТОВ И ПРИЛОЖЕНИЙ	93
6.1.	Идентификаторы объектов	93
6.2.	Идентификаторы приложения	94

1. Введение

В настоящем приложении уточняются форматы данных, элементы данных и структуры данных, подлежащие использованию в записывающем оборудовании и карточках тахографа.

1.1. Метод определения типов данных

Для определения типов данных в настоящем приложении используется абстрактное описание синтаксиса версии 1 (ASN.1). Эта система позволяет определить простые и структурированные данные, не прибегая к помощи какого-либо конкретного синтаксиса передачи (правил кодирования), который зависит от приложения и операционной среды.

Правила присвоения названий типа ASN.1 соответствуют стандарту ISO/IEC 88241. Это предполагает, что:

- при возможности смысл соответствующего типа данных косвенно заложен в выбранных названиях,
- если какой-либо тип данных состоит из других типов данных, название этого типа данных и в этом случае представляет собой простую последовательность буквенных знаков, которая начинается с заглавной буквы; вместе с тем заглавные буквы используются и в названии с целью придать данным соответствующий смысл,
- в целом, название типов данных соотносится с названием тех типов данных, с помощью которых они построены, с оборудованием, в которых хранятся данные, и с функцией, имеющей отношение к данным.

Если какой-либо тип ASN.1 уже определён в качестве того или иного стандарта и если он подходит для использования в записывающем оборудовании, в данном приложении будет определён и этот тип ASN.1.

Для того, чтобы можно было использовать несколько типов правил кодирования, некоторые типы ASN.1 в настоящем приложении ограничиваются соответствующими идентификаторами диапазона значений. Идентификаторы диапазона значений определяются в пункте 3 и приложении 2.

1.2. Ссылки

В настоящем приложении используются следующие источники:

ISO 639	Код отображения названий языков. 1-е издание: 1988.
ISO 3166	Коды отображения названий стран и их частей. Часть 1: коды стран, 2013
ISO 3779	Дорожные транспортные средства. Опознавательный номер транспортного средства (VIN). Содержание и структура. 2009
ISO/IEC 7816-5	Идентификационные карточки – карточки с интегральными микросхемами. Часть 5: регистрация подателей заявок. 2-е издание: 2004.
ISO/IEC 7816-6	Идентификационные карточки – карточки с интегральными микросхемами. Часть 6: межотраслевые элементы данных для обмена, 2004 + техническая поправка 1: 2006
ISO/IEC 8824-1	Информационные технологии. Абстрактное описание синтаксиса версии 1 (ASN.1): спецификация базового описания. 2008 + техническая поправка 1: 2012 и техническая поправка 2: 2014.
ISO/IEC 8825-2	Информационные технологии. Правила кодирования ASN.1: спецификация правил компактного кодирования (PER). 2008.
ISO/IEC 8859-1	Информационные технологии. 8-битные однобайтовые кодированные наборы графических символов. Часть 1: латиница № 1. 1-е издание: 1998.
ISO/IEC 8859-7	Информационные технологии. 8-битные однобайтовые кодированные наборы графических символов. Часть 7: латиница/греческий алфавит. 2003.
ISO 16844-3	Дорожные транспортные средства. Системы тахографов. Интерфейс датчика движения. 2004 + техническая поправка 1: 2006.
TR-03110-3	Технические руководящие принципы BSI/ANSSI TR-03110-3, механизмы повышенной безопасности для машиночитываемых проездных документов и токен eIDAS. Часть 3: общие спецификации, версия 2.20, 3. Февраль 2015 г.

2. Определения типов данных

В случае любого из следующих типов данных значение по умолчанию содержания «нет данных» или «неприменимо» определяется посредством заполнения соответствующего элемента данных с помощью байтов «FF».

Все типы данных используются для приложений первого и второго поколений, если не указано иначе.

2.1. ActivityChangeInfo

Этот тип данных позволяет кодировать с помощью слова из двух байтов состояние считывающего устройства в 00:00 часов и/или статус водителя в 00:00 часов и/или изменения вида деятельности и/или изменения статуса управления и/или изменения статуса карточки водителя или второго водителя. Данный тип данных связан с требованиями 105, 266, 291, 320, 321, 343 и 344 в приложении 1С.

ActivityChangeInfo ::= OCTET STRING (SIZE(2))

Присвоение значения – выровненный октет: 's'В (16 бит)

Для записи данных в блок памяти (или состояние считывающего устройства):

's'В	Считывающее устройство: '0'В: DRIVER, '1'В: CO-DRIVER,
'c'В	Статус управления: '0'В: SINGLE, '1'В: CREW,
'p'В устройстве:	Статус карточки водителя (или мастерской) в соответствующем считывающем устройстве: '0'В: INSERTED, карточка вставлена, '1'В: NOT INSERTED, карточка не вставлена (или извлечена),
'aa'В	Деятельность: '00'В: BREAK/REST, '01'В: AVAILABILITY, '10'В: WORK, '11'В: DRIVING,
'tttttttt'В	Время изменения: число минут, начиная с 00:00 часов на данный день.

Для записи данных на карточку водителя (или мастерской) (и статуса водителя):

's'В	Считывающее устройство (неактуально, если 'p'=1, за исключением примечания ниже): '0'В: DRIVER, '1'В: CO-DRIVER,
'c'В (случай 'p'=1):	Статус управления (случай 'p'=0) или Статус следующего вида деятельности '0'В: SINGLE, '0'В: UNKNOWN '1'В: CREW, '1'В: KNOWN (= введено вручную)
'p'В	Статус карточки: '0'В: INSERTED, карточка вставлена в записывающее оборудование, '1'В: NOT INSERTED, карточка не вставлена (или извлечена),
'aa'В	Деятельность (неактуально, если 'p'=1 и 'c'=0, за исключением примечания ниже): '00'В: BREAK/REST, '01'В: AVAILABILITY, '10'В: WORK, '11'В: DRIVING,
'tttttttt'В	Время изменения: число минут, начиная с 00:00 часов на данный день.

Примечание на случай «извлечения карточки»:

Когда карточка извлечена:

- знак 's' применим и указывает на считывающее устройство, из которого извлечена карточка,
- 'c' должно быть установлено на 0,
- 'p' должно быть установлено на 1,
- 'aa' должно кодировать текущий вид деятельности, выбранный в указанное время,

В результате ручного ввода биты 'c' и 'aa' в составе слова (хранящиеся в памяти карточки) позднее могут быть удалены и на их место записаны другие данные, отражающие факт этого ввода.

2.2. Address

Адрес.

```
Address ::= SEQUENCE {
    codePage          INTEGER (0..255),
    address           OCTET STRING (SIZE(35))
}
```

codePage определяет набор символов, указанный в главе 4,

address представляет собой адрес, закодированный с помощью указанного набора символов.

2.3. AESKey

Второе поколение:

Ключ AES длиной 128, 192 или 256 бит.

```
AESKey ::= CHOICE {
    aes128Key          AES128Key,
    aes192Key          AES192Key,
    aes256Key          AES256Key
}
```

Присвоение значения: дополнительно не указывается.

2.4. AES128Key

Второе поколение:

Ключ AES128.

```
AES128Key ::= SEQUENCE {
    length             INTEGER(0..255),
    aes128Key          OCTET STRING (SIZE(16))
}
```

length обозначает октетную длину ключа AES128.

aes128Key – ключ AES длиной 128 бит.

Присвоение значения:

значение длины – 16.

2.5. AES192Key

Второе поколение:

Ключ AES192.

```
AES192Key ::= SEQUENCE {
    length             INTEGER(0..255),
    aes192Key          OCTET STRING (SIZE(24))
}
```

length обозначает октетную длину ключа AES192.

aes192Key – ключ AES длиной 192 бит.

Присвоение значения:

значение длины – 24.

2.6. AES256Key

Второе поколение:

Ключ AES256.

```
AES256Key ::= SEQUENCE {  
    length                INTEGER(0..255),  
    aes256Key             OCTET STRING (SIZE(32))  
}
```

length обозначает октетную длину ключа AES256.

aes256Key – ключ AES длиной 256 бит.

Присвоение значения:

значение длины – 32.

2.7. BCDString

BCDString используется для отображения десятичного числа в двоичном коде (BCD). Этот тип данных используется для отображения одного десятичного знака в полуоктете (4 бита). BCDString определяется в соответствии со стандартом ISO/IEC 8824-1 CharacterStringType.

```
BCDString ::= CHARACTER STRING (WITH COMPONENTS {  
    identification ( WITH COMPONENTS {  
        fixed PRESENT }) })
```

BCDString использует нотацию «hstring». Крайняя левая шестнадцатеричная цифра представляет собой самый значимый полуоктет первого октета. Для получения нескольких октетов после крайнего левого полуоктета первого октета по мере необходимости включается соответствующее число нулевых полуоктетов.

Допустимые цифры: 0, 1, .. 9.

2.8. CalibrationPurpose

Данный код указывает причину регистрации набора параметров калибровки. Данный тип данных связан с требованиями 097 и 098 приложения 1B и требованием 119 приложения 1C.

```
CalibrationPurpose ::= OCTET STRING (SIZE(1))
```

Присвоение значения:

Первое поколение:

- '00'H зарезервированное значение,
- '01'H активация: регистрация известных параметров калибровки в момент включения БУ,
- '02'H первая установка: первая калибровка БУ после его активации,
- '03'H установка: первая калибровка БУ на текущем транспортном средстве,
- '04'H регулярная проверка.

Второе поколение:

В дополнение к первому поколению, используются следующие значения:

- '05'H ввод VRN предприятием,
- '06'H корректировка времени без калибровки,
- от '07'H до '7F'H RFU, от '80'H до 'FF'H зависит от производителя.

2.9. CardActivityDailyRecord

Информация, которая хранится на карточке, относится к деятельности водителя за конкретный календарный день. Данный тип данных связан с требованиями 266, 291, 320 и 343 в приложении 1С.

```
CardActivityDailyRecord ::= SEQUENCE {
    activityPreviousRecordLength    INTEGER(0..CardActivityLengthRange),
    activityRecordLength            INTEGER(0..CardActivityLengthRange),
    activityRecordDate              TimeReal,
    activityDailyPresenceCounter    DailyPresenceCounter,
    activityDayDistance             Distance,
    activityChangeInfo              SET SIZE(1..1440) OF ActivityChangeInfo
}
```

activityPreviousRecordLength – общая длина предыдущей ежедневной записи в байтах. Максимальное значение определяется в виде длины ОКТЕТНОЙ СТРОКИ, содержащей эти записи (см. CardActivityLengthRange приложение 2, пункт 4). Когда эта запись становится самой старой ежедневной записью, значение activityPreviousRecordLength должно устанавливаться на 0.

activityRecordLength – общая длина данной записи в байтах. Максимальное значение определяется в виде длины ОКТЕТНОЙ СТРОКИ, содержащей эти записи.

activityRecordDate – дата записи.

activityDailyPresenceCounter – счётчик ежедневного наличия карточки на данный день.

activityDayDistance – общее расстояние, пройденное за данный день.

activityChangeInfo – набор данных ActivityChangeInfo в отношении водителя на данный день. Он может содержать максимум 1 440 значений (изменение вида деятельности 1 раз в минуту). Этот набор данных всегда включает кодирование статуса водителя activityChangeInfo на 00:00 часов.

2.10. CardActivityLengthRange

Число байтов на карточке водителя или мастерской, которые предусмотрены для хранения записей, касающихся деятельности водителя.

```
CardActivityLengthRange ::= INTEGER(0..216-1)
```

Присвоение значения: см. приложение 2.

2.11. CardApprovalNumber

Номер официального утверждения типа карточки.

```
CardApprovalNumber ::= IA5String(SIZE(8))
```

Присвоение значения:

Номер официального утверждения указывается в соответствии с публикацией на соответствующем веб-сайте Европейской комиссии, например, с дефисами, если они есть. Номер официального утверждения смещён влево.

2.12. CardCertificate

Первое поколение:

Сертификат открытого ключа карточки.

```
CardCertificate ::= Certificate
```

2.13. CardChipIdentification

Информация, записанная на карточке, которая относится к идентификации интегральной схемы карточки (ИС) (требование 249 приложения 1С). icSerialNumber вместе с icManufacturingReferences представляют собой уникальный определитель микросхемы карточки. Один только icSerialNumber не является уникальным определителем микросхемы карточки.

```
CardChipIdentification ::= SEQUENCE {
    icSerialNumber          OCTET STRING (SIZE(4)),
    icManufacturingReferences OCTET STRING (SIZE(4))
}
```

icSerialNumber – серийный номер ИС.

icManufacturingReferences – конкретный идентификатор производителя ИС.

2.14. CardConsecutiveIndex

Порядковый индекс карточки (определение h)).

```
CardConsecutiveIndex ::= IA5String(SIZE(1))
```

Присвоение значения: (см. приложение 1С, глава 7)

Порядок повышения: ‘0, ..., 9, A, ..., Z, a, ..., z’

2.15. CardControlActivityDataRecord

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к последней проверке, которой подвергался водитель (приложение 1С, требования 274, 299, 327 и 350).

```
CardControlActivityDataRecord ::= SEQUENCE {
    controlType          ControlType,
    controlTime          TimeReal,
    controlCardNumber   FullCardNumber,
    controlVehicleRegistration VehicleRegistrationIdentification,
    controlDownloadPeriodBegin TimeReal,
    controlDownloadPeriodEnd TimeReal
}
```

controlType – тип контроля.

controlTime – дата и время контроля.

controlCardNumber – FullCardNumber контролёра, проводившего проверку.

controlVehicleRegistration – VRN и государство-член регистрации транспортного средства, к которому относится контроль.

controlDownloadPeriodBegin и **controlDownloadPeriodEnd** – период, за который загружаются данные, в случае загрузки.

2.16. CardCurrentUse

Информация о фактическом пользовании карточкой (приложение 1С, требования 273, 298, 326 и 349).

```
CardCurrentUse ::= SEQUENCE {
    sessionOpenTime          TimeReal,
    sessionOpenVehicle       VehicleRegistrationIdentification
}
```

sessionOpenTime – время, когда карточка была вставлена для текущего использования. При извлечении карточки данный элемент устанавливается на ноль.

sessionOpenVehicle – идентификация текущего используемого транспортного средства, установленная при вводе карточки. При извлечении карточки данный элемент устанавливается на ноль.

2.17. CardDriverActivity

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к деятельности водителя (приложение 1С, требования 267, 268, 292, 293, 321 и 344).

```
CardDriverActivity ::= SEQUENCE {
    activityPointerOldestDayRecord    INTEGER(0.. CardActivityLengthRange-1),
    activityPointerNewestRecord       INTEGER(0.. CardActivityLengthRange-1),
    activityDailyRecords              OCTET STRING
                                     (SIZE(CardActivityLengthRange))
}
```

activityPointerOldestDayRecord – спецификация начала места хранения (число байтов с начала строки) самой давней записи о полном дне в строке activityDailyRecords. Максимальное значение определяется в виде длины строки.

activityPointerNewestRecord – спецификация начала места хранения (число байтов с начала строки) самой новой дневной записи в строке activityDailyRecords. Максимальное значение определяется в виде длины строки.

activityDailyRecords – место для хранения данных о деятельности водителя (структура данных: CardActivityDailyRecord) за каждый календарный день, когда использовалась карточка.

Присвоение значения: данная октетная строка периодически заполняется записями типа CardActivityDailyRecord. При первом использовании хранение данных производится с начала первого байта строки. Все новые записи включаются в конце предыдущей. Когда вся строка заполняется, процесс хранения продолжается с первого байта строки, независимо от наличия разрыва в том или ином элементе данных. До включения в строку данных о новом виде деятельности (посредством расширения текущей позиции activityDailyRecord или включения новой позиции activityDailyRecord), которые записываются вместо прежних данных о деятельности, указатель activityPointerOldestDayRecord должен быть обновлён с целью отразить новое место хранения самой давней полной дневной записи, а указатель activityPreviousRecordLength этой (новой) самой давней полной дневной записи должен быть установлен на ноль.

2.18. CardDrivingLicenceInformation

Информация, записанная на карточке водителя, которая относится к данным о водительском удостоверении владельца карточки (приложение 1С, требования 259 и 284).

```
CardDrivingLicenceInformation ::= SEQUENCE {
    drivingLicenceIssuingAuthority    Name,
    drivingLicenceIssuingNation       NationNumeric,
    drivingLicenceNumber              IA5String(SIZE(16))
}
```

drivingLicenceIssuingAuthority – орган, ответственный за выдачу водительского удостоверения.

drivingLicenceIssuingNation – национальная принадлежность органа, выдавшего водительское удостоверение.

drivingLicenceNumber – номер водительского удостоверения.

2.19. CardEventData

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к событиям, связанным с владельцем карточки (приложение 1С, требования 260, 285, 318 и 341).

```
CardEventData ::= SEQUENCE SIZE(6) OF {
    cardEventRecords                  SET SIZE(NoOfEventsPerType) OF
                                     CardEventRecord
}
```

CardEventData – последовательность записей cardEventRecords, записанная в порядке возрастания значения элемента EventFaultType (за исключением записей, касающихся нарушения защиты, которые группируются в последнем массиве данных последовательности).

cardEventRecords – набор записей о событиях данного типа (или категория событий, имеющих отношение к попыткам нарушения защиты).

2.20. CardEventRecord

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к событию, связанному с владельцем карточки (приложение 1С, требования 261, 286, 318 и 341).

```
CardEventRecord ::= SEQUENCE {
    eventType                EventFaultType,
    eventBeginTime           TimeReal,
    eventEndTime             TimeReal,
    eventVehicleRegistration VehicleRegistrationIdentification
}
```

eventType – тип события.

eventBeginTime – дата и время начала события.

eventEndTime – дата и время завершения события.

eventVehicleRegistration – VRN и государство-член регистрации транспортного средства, к которому относится событие.

2.21. CardFaultData

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к неисправностям, связанным с владельцем карточки (приложение 1С, требования 263, 288, 318 и 341).

```
CardFaultData ::= SEQUENCE SIZE(2) OF {
    cardFaultRecords        SET SIZE(NoOfFaultsPerType) OF
                            CardFaultRecord
}
```

CardFaultData – последовательность совокупности записей, отражающих неисправности записывающего оборудования, за которой следует совокупность записей, отражающих сбой в работе карточек.

cardFaultRecords – совокупность записей о неисправностях, сгруппированных по данной категории неисправностей (записывающего оборудования или карточки).

2.22. CardFaultRecord

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к неисправности, связанной с владельцем карточки (приложение 1С, требования 264, 289, 318 и 341).

```
CardFaultRecord ::= SEQUENCE {
    faultType                EventFaultType,
    faultBeginTime           TimeReal,
    faultEndTime             TimeReal,
    faultVehicleRegistration VehicleRegistrationIdentification
}
```

faultType – тип неисправности.

faultBeginTime – дата и время начала неисправности.

faultEndTime – дата и время окончания неисправности.

faultVehicleRegistration – VRN и государство-член регистрации транспортного средства, к которому относится неисправность.

2.23. CardIccIdentification

Информация, записанная на карточке, которая относится к идентификации интегральной схемы карточки (ИС) (требование 248 приложения 1С).

```
CardIccIdentification ::= SEQUENCE {
    clockStop                OCTET STRING (SIZE(1)),
    cardExtendedSerialNumber ExtendedSerialNumber,
    cardApprovalNumber       CardApprovalNumber,
    cardPersonaliserID       ManufacturerCode,
    embedderIcAssemblerId    EmbedderIcAssemblerId,
    icIdentifier              OCTET STRING (SIZE(2))
}
```

clockStop – режим остановки часов, определённый в приложении 2.

cardExtendedSerialNumber – уникальный серийный номер карточки с ИС, как указано далее для типа данных ExtendedSerialNumber.

cardApprovalNumber – номер официального утверждения типа карточки.

cardPersonaliserID – идентификационные данные персонализатора карточки, закодированные как ManufacturerCode.

embedderIcAssemblerId предоставляет информацию об установщике/сборщике ИС.

icIdentifier – идентификатор ИС на карточке и производителя ИС, как указано в стандарте ISO/IEC 7816-6.

2.24. CardIdentification

Информация, записанная на карточке, которая имеет отношение к идентификационным данным карточки (приложение 1С, требования 255, 280, 310, 333, 359, 365, 371 и 377).

```
CardIdentification ::= SEQUENCE {
    cardIssuingMemberState  NationNumeric,
    cardNumber              CardNumber,
    cardIssuingAuthorityName Name,
    cardIssueDate           TimeReal,
    cardValidityBegin       TimeReal,
    cardExpiryDate          TimeReal
}
```

cardIssuingMemberState – код государства-члена, выдавшего карточку.

cardNumber – номер карточки.

cardIssuingAuthorityName – название органа, выдавшего карточку.

cardIssueDate – дата выдачи карточки нынешнему владельцу.

cardValidityBegin – первая дата действия карточки.

cardExpiryDate – дата истечения срока действия карточки.

2.25. CardMACertificate

Второе поколение:

Сертификат открытого ключа карточки для взаимной проверки подлинности с БУ. Структура данного сертификата представлена в приложении 11.

```
CardMACertificate ::= Certificate
```

2.26. CardNumber

Номер карточки в соответствии с определением g.

```
CardNumber ::= CHOICE {
  SEQUENCE {
    driverIdentification      IA5String(SIZE(14)),
    cardReplacementIndex     CardReplacementIndex,
    cardRenewalIndex         CardRenewalIndex
  },
  SEQUENCE {
    ownerIdentification      IA5String(SIZE(13)),
    cardConsecutiveIndex     CardConsecutiveIndex,
    cardReplacementIndex     CardReplacementIndex,
    cardRenewalIndex         CardRenewalIndex
  }
}
```

driverIdentification – уникальная идентификация водителя в государстве-члене.

ownerIdentification – уникальная идентификация предприятия или мастерской или контрольного органа в государстве-члене.

cardConsecutiveIndex – порядковый индекс карточки.

cardReplacementIndex – индекс замены карточки.

cardRenewalIndex – индекс возобновления карточки.

Первая последовательность выбора подходит для кодирования номера карточки водителя, вторая последовательность выбора подходит для кодирования номеров карточки мастерской, контрольной карточки и карточки предприятия.

2.27. CardPlaceDailyWorkPeriod

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к местам, где начинаются и/или заканчиваются дневные периоды работы (приложение 1С, требования 272, 297, 325 и 348).

```
CardPlaceDailyWorkPeriod ::= SEQUENCE {
  placePointerNewestRecord INTEGER(0 .. NoOfCardPlaceRecords-1),
  placeRecords              SET SIZE(NoOfCardPlaceRecords) OF PlaceRecord
}
```

placePointerNewestRecord – индекс последней обновлённой записи данных о месте.

Присвоение значения: Число, соответствующее числовому показателю записи данных о месте, которое начинается с 0 в случае первой регистрации записей, касающихся места, в структуре.

placeRecords – совокупность записей, содержащих информацию о введённых названиях мест.

2.28. CardPrivateKey

Первое поколение:

Закрытый ключ карточки.

```
CardPrivateKey ::= RSAKeyPrivateExponent
```

2.29. CardPublicKey

Открытый ключ карточки.

```
CardPublicKey ::= PublicKey
```

2.30. CardRenewalIndex

Индекс возобновления карточки (определение i).

```
CardRenewalIndex ::= IA5String(SIZE(1))
```

Присвоение значения: (см. главу VII данного приложения).

‘0’ Первый выпуск.

Порядок повышения: ‘0, ..., 9, A, ..., Z’

2.31. CardReplacementIndex

Индекс замены карточки (определение j).

```
CardReplacementIndex ::= IA5String(SIZE(1))
```

Присвоение значения: (см. главу VII данного приложения).

‘0’ Оригинальная карточка.

Порядок повышения: ‘0, ..., 9, A, ..., Z’

2.32. CardSignCertificate

Второе поколение:

Сертификат открытого ключа карточки для подписания. Структура данного сертификата представлена в приложении 11.

```
CardSignCertificate ::= Certificate
```

2.33. CardSlotNumber

Код, позволяющий проводить различие между двумя считывающими устройствами бортового устройства.

```
CardSlotNumber ::= INTEGER {  
    driverSlot                (0),  
    co-driverSlot             (1)  
}
```

Присвоение значения: дополнительно не указывается.

2.34. CardSlotsStatus

Код, указывающий тип карточек, вставленных в два считывающих устройства бортового устройства.

```
CardSlotsStatus ::= OCTET STRING (SIZE(1))
```

Присвоение значения – выровненный октет: ‘ccccddd’В

‘cccc’В Идентификационные данные типа карточки, вставленной в считывающее устройство второго водителя,

‘ddd’В Идентификационные данные типа карточки, вставленной в считывающее устройство водителя,

со следующими идентификационными кодами:

‘0000’В карточка не вставлена,

‘0001’В вставлена карточка водителя,

‘0010’В вставлена карточка мастерской,

‘0011’В вставлена контрольная карточка,

‘0100’В вставлена карточка предприятия.

2.35. CardSlotsStatusRecordArray

Второе поколение:

CardSlotsStatus и метаданные, используемые в протоколе загрузки.

```

CardSlotsStatusRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize         INTEGER(1..65535),
    noOfRecords        INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF CardSlotsStatus
}

```

recordType указывает на тип записи (CardSlotsStatus). **Присвоение значения:** см. RecordType

recordSize – размер CardSlotsStatus в байтах.

noOfRecords – число записей в массиве.

records – массив записей CardSlotsStatus.

2.36. CardStructureVersion

Код, указывающий на версию реализованной структуры на карточке тахографа.

```

CardStructureVersion ::= OCTET STRING (SIZE(2))

```

Присвоение значения: ‘aabb’H:

‘aa’H индекс изменений структуры.
‘00’H для приложений первого поколения
‘01’H для приложений второго поколения

‘bb’H индекс изменений, касающийся использования элементов данных, определённых в структуре, заданной стартовым байтом.
‘00’H для данной версии приложений первого поколения
‘00’H для данной версии приложений второго поколения

2.37. CardVehicleRecord

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к периоду пользования транспортным средством в течение календарного дня (приложение 1С, требования 269, 294, 322 и 345).

Первое поколение:

```

CardVehicleRecord ::= SEQUENCE {
    vehicleOdometerBegin      OdometerShort,
    vehicleOdometerEnd       OdometerShort,
    vehicleFirstUse          TimeReal,
    vehicleLastUse           TimeReal,
    vehicleRegistration       VehicleRegistrationIdentification,
    vuDataBlockCounter       VuDataBlockCounter
}

```

vehicleOdometerBegin – показания одометра транспортного средства на начало периода эксплуатации транспортного средства.

vehicleOdometerEnd – показания одометра транспортного средства на конец периода эксплуатации транспортного средства.

vehicleFirstUse – дата и время начала периода эксплуатации транспортного средства.

vehicleLastUse – дата и время окончания периода эксплуатации транспортного средства.

vehicleRegistration – VRN и государство-член регистрации транспортного средства.

vuDataBlockCounter – значение VuDataBlockCounter на момент последнего извлечения периода эксплуатации транспортного средства.

Второе поколение:

```
CardVehicleRecord ::= SEQUENCE {
    vehicleOdometerBegin          OdometerShort,
    vehicleOdometerEnd           OdometerShort,
    vehicleFirstUse              TimeReal,
    vehicleLastUse               TimeReal,
    vehicleRegistration           VehicleRegistrationIdentification,
    vuDataBlockCounter           VuDataBlockCounter,
    vehicleIdentificationNumber   VehicleIdentificationNumber
}
```

В дополнение к первому поколению, используется следующий элемент данных:

VehicleIdentificationNumber – идентификационный номер транспортного средства, относящийся ко всему транспортному средству.

2.38. CardVehiclesUsed

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к транспортным средствам, эксплуатируемым владельцем карточки (приложение 1С, требования 270, 295, 323 и 346).

```
CardVehiclesUsed ::= SEQUENCE {
    vehiclePointerNewestRecord    INTEGER(0..NoOfCardVehicleRecords-1),
    cardVehicleRecords           SET SIZE(NoOfCardVehicleRecords) OF
                                CardVehicleRecord
}
```

vehiclePointerNewestRecord – индекс последней обновлённой записи, касающейся транспортного средства.

Присвоение значения: Число, соответствующее числовому показателю записи данных о транспортном средстве, которое начинается с 0 в случае первой регистрации записей, касающихся транспортного средства, в структуре.

cardVehicleRecords – совокупность записей, содержащих информацию об использованных транспортных средствах.

2.39. CardVehicleUnitRecord

Второе поколение:

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к используемым бортовым устройствам (приложение 1С, требования 303 и 351).

```
CardVehicleUnitRecord ::= SEQUENCE {
    timeStamp                    TimeReal,
    manufacturerCode             ManufacturerCode,
    deviceID                     INTEGER(0..255),
    vuSoftwareVersion            VuSoftwareVersion
}
```

timeStamp – начало периода использования бортового устройства (т.е. первый ввод карточки в бортовое устройство в течение рассматриваемого периода).

manufacturerCode идентифицирует производителя бортового устройства.

deviceID идентифицирует тип бортового устройства производителя. Значение зависит от производителя.

vuSoftwareVersion – номер версии программного обеспечения бортового устройства.

2.40. CardVehicleUnitsUsed

Второе поколение:

Информация, записанная на карточке водителя или мастерской, которая имеет отношение к бортовым устройствам, используемым владельцем карточки (приложение 1С, требования 306 и 352).

```
CardVehicleUnitsUsed := SEQUENCE {  
    vehicleUnitPointerNewestRecord    INTEGER(0..NoOfCardVehicleUnitRecords-  
                                        1),  
    cardVehicleUnitRecords            SET SIZE(NoOfCardVehicleUnitRecords) OF  
                                        CardVehicleUnitRecord  
}
```

vehicleUnitPointerNewestRecord – индекс последней обновлённой записи, касающейся бортового устройства.

Присвоение значения: Число, соответствующее числовому показателю записи данных о бортовом устройстве, которое начинается с 0 в случае первой регистрации записей, касающихся бортового устройства, в структуре.

cardVehicleUnitRecords – совокупность записей, содержащих информацию об использованных бортовых устройствах.

2.41. Certificate

Сертификат открытого ключа, выданный сертификационным органом.

Первое поколение:

```
Certificate ::= OCTET STRING (SIZE(194))
```

Присвоение значения: цифровая подпись с частичным восстановлением CertificateContent в соответствии с общими механизмами защиты, описанными в приложении 11: Подпись (128 байт) || Остаток открытого ключа (58 байт) || Код сертификационного органа (8 байт).

Второе поколение:

```
Certificate ::= OCTET STRING (SIZE(204..341))
```

Присвоение значения: См. приложение 11

2.42. CertificateContent

Первое поколение:

(Ясное) содержание сертификата открытого ключа в соответствии с общими механизмами защиты, изложенными в приложении 11.

```
CertificateContent ::= SEQUENCE {
    certificateProfileIdentifier    INTEGER(0..255),
    certificationAuthorityReference KeyIdentifier,
    certificateHolderAuthorisation CertificateHolderAuthorisation,
    certificateEndOfValidity       TimeReal,
    certificateHolderReference      KeyIdentifier,
    publicKey                      PublicKey
}
```

certificateProfileIdentifier – версия соответствующего сертификата.

Присвоение значения: '01h' для данной версии.

certificationAuthorityReference идентифицирует сертификационный орган, выдавший сертификат. Также включает в себя ссылку на открытый ключ данного сертификационного органа.

certificateHolderAuthorisation – идентификатор прав держателя сертификата.

certificateEndOfValidity – дата, когда истекает срок административного действия сертификата.

certificateHolderReference – идентификатор держателя сертификата. Также включает в себя ссылку на его открытый ключ.

publicKey – открытый ключ, подтверждающий данный сертификат.

2.43. CertificateHolderAuthorisation

Идентификатор прав держателя сертификата.

```
CertificateHolderAuthorisation ::= SEQUENCE {
    tachographApplicationID    OCTET STRING(SIZE(6))
    equipmentType              EquipmentType
}
```

Первое поколение:

tachographApplicationID – идентификатор приложения для прикладной программы тахографа.

Присвоение значения: 'FFh' '54h' '41h' '43h' '48h' '4Fh'. Этот AID является фирменным незарегистрированным идентификатором приложения в соответствии со стандартом ISO/IEC 7816-5.

equipmentType – идентификатор типа оборудования, для которого предназначен сертификат.

Присвоение значения: в соответствии с типом данных EquipmentType. **0**, если это сертификат государства-члена.

Второе поколение:

tachographApplicationID обозначает 6 наиболее значимых байтов идентификатора (AID) приложения карточки тахографа второго поколения. AID приложения карточки тахографа указан в главе 6.2.

Присвоение значения: 'FF 53 4D 52 44 54'.

equipmentType – идентификатор типа оборудования второго поколения, для которого предназначен сертификат.

Присвоение значения: в соответствии с типом данных EquipmentType.

2.44. CertificateRequestID

Уникальный идентификатор заявки на получение сертификата. Также может использоваться в качестве идентификатора открытого ключа бортового устройства, если серийный номер бортового устройства, для которого предназначен данный ключ, в момент создания сертификата неизвестен.

```
CertificateRequestID ::= SEQUENCE{
    requestSerialNumber      INTEGER(0..232-1),
    requestMonthYear         BCDString(SIZE(2)),
    crIdentifier              OCTET STRING(SIZE(1)),
    manufacturerCode         ManufacturerCode
}
```

requestSerialNumber – серийный номер заявки на сертификат, уникальный для данного производителя и относящийся к месяцу, указанному ниже.

requestMonthYear – идентификатор месяца и года заявки на сертификат.

Присвоение значения: код BCD месяца (две цифры) и года (две последние цифры).

crIdentifier – идентификатор, позволяющий проводить различие между заявкой на сертификат и расширенным серийным номером.

Присвоение значения: 'FFh'.

manufacturerCode – цифровой код производителя, подающего заявку на сертификат.

2.45. CertificationAuthorityKID

Идентификатор открытого ключа сертификационного органа (государства-члена или европейского сертификационного органа).

```
CertificationAuthorityKID ::= SEQUENCE{
    nationNumeric             NationNumeric,
    nationAlpha               NationAlpha,
    keySerialNumber          INTEGER(0..255),
    additionalInfo           OCTET STRING(SIZE(2)),
    caIdentifier              OCTET STRING(SIZE(1))
}
```

nationNumeric – цифровой код страны сертификационного органа.

nationAlpha – буквенно-цифровой код страны сертификационного органа.

keySerialNumber – серийный номер, позволяющий проводить различие между различными ключами сертификационного органа в случае изменения ключей.

additionalInfo – двухбайтовое поле для дополнительного кодирования (специально для сертификационного органа).

caIdentifier – идентификатор, позволяющий проводить различие между идентификатором ключа сертификационного органа и другими идентификаторами ключа.

Присвоение значения: '01h'.

2.46. CompanyActivityData

Информация, записанная на карточке предприятия, которая относится к действиям, проводимым с карточкой (приложение 1С, требования 373 и 379).

```
CompanyActivityData ::= SEQUENCE {
  companyPointerNewestRecord    INTEGER (0..NoOfCompanyActivityRecords-1),
  companyActivityRecords       SET SIZE (NoOfCompanyActivityRecords) OF
  companyActivityRecord        SEQUENCE {
    companyActivityType         CompanyActivityType,
    companyActivityTime         TimeReal,
    cardNumberInformation       FullCardNumber,
    vehicleRegistrationInformation VehicleRegistrationIdentification,
    downloadPeriodBegin         TimeReal,
    downloadPeriodEnd           TimeReal
  }
}
```

companyPointerNewestRecord – индекс последней обновлённой записи companyActivityRecord.

Присвоение значения: Число, соответствующее числовому показателю записи данных о деятельности предприятия, которое начинается с 0 в случае первой регистрации записей, касающихся деятельности предприятия, в структуре.

companyActivityRecords – совокупность всех записей действий, произведённых предприятием.

companyActivityRecord – последовательность информации, относящейся к одному действию, произведённому предприятием.

companyActivityType – тип действия предприятия.

companyActivityTime – дата и время действия предприятия.

cardNumberInformation – номер карточки и название государства-члена, выдавшего карточку с загруженными с неё данными, если применимо.

vehicleRegistrationInformation – VRN и государство-член регистрации транспортного средства, в отношении которого данные загружены, заблокированы или разблокированы.

downloadPeriodBegin и **downloadPeriodEnd** – период, за который в соответствующих случаях загружены данные с БУ.

2.47. CompanyActivityType

Код указывающей на действия, произведённые предприятием с использованием карточки предприятия.

```
CompanyActivityType ::= INTEGER {
  card downloading           (1),
  VU downloading            (2),
  VU lock-in                 (3),
  VU lock-out                (4)
}
```

2.48. CompanyCardApplicationIdentification

Информация, записанная на карточке предприятия, которая относится к идентификации приложения карточки (приложение 1С, требования 369 и 375).

```
CompanyCardApplicationIdentification ::= SEQUENCE {
  typeOfTachographCardId     EquipmentType,
  cardStructureVersion        CardStructureVersion,
  noOfCompanyActivityRecords  NoOfCompanyActivityRecords
}
```

typeOfTachographCardId уточняет реализованный тип карточки.

cardStructureVersion уточняет версию структуры, реализованной на карточке.

noOfCompanyActivityRecords – число записей о действиях предприятия, которые могут храниться на карточке.

2.49. CompanyCardHolderIdentification

Информация, записанная на карточке предприятия, которая относится к идентификации владельца карточки (приложение 1С, требования 372 и 378).

```
CompanyCardHolderIdentification ::= SEQUENCE {
    companyName                Name,
    companyAddress              Address,
    cardHolderPreferredLanguage Language
}
```

companyName – название предприятия владельца.

companyAddress – адрес предприятия владельца.

cardHolderPreferredLanguage – предпочитаемый язык владельца карточки.

2.50. ControlCardApplicationIdentification

Информация, записанная на контрольной карточке, которая относится к идентификации приложения карточки (приложение 1С, требования 357 и 363).

```
ControlCardApplicationIdentification ::= SEQUENCE {
    typeOfTachographCardId      EquipmentType,
    cardStructureVersion         CardStructureVersion,
    noOfControlActivityRecords   NoOfControlActivityRecords
}
```

typeOfTachographCardId уточняет реализованный тип карточки.

cardStructureVersion уточняет версию структуры, реализованной на карточке.

noOfControlActivityRecords – число записей о контрольных действиях, которые могут храниться на карточке.

2.51. ControlCardControlActivityData

Информация, записанная на контрольной карточке, которая относится к контрольным действиям, проводимым с карточкой (приложение 1С, требования 361 и 367).

```
ControlCardControlActivityData ::= SEQUENCE {
    controlPointerNewestRecord    INTEGER(0.. NoOfControlActivityRecords-1),
    controlActivityRecords        SET SIZE(NoOfControlActivityRecords) OF
        controlActivityRecord     SEQUENCE {
            controlType            ControlType,
            controlTime            TimeReal,
            controlledCardNumber   FullCardNumber,
            controlledVehicleRegistration VehicleRegistrationIdentification,
            controlDownloadPeriodBegin TimeReal,
            controlDownloadPeriodEnd TimeReal
        }
}
```

controlPointerNewestRecord – индекс последней обновлённой записи, касающейся контрольных действий.

Присвоение значения: Число, соответствующее числовому показателю записи данных о контрольной деятельности, которое начинается с 0 в случае первой регистрации записей, касающихся контрольной деятельности, в структуре.

controlActivityRecords – совокупность всех записей контрольных действий.

controlActivityRecord – последовательность информации, относящейся к одному контрольному действию.

controlType – тип контроля.

controlTime – дата и время контроля.

controlledCardNumber – номер карточки и название государства-члена, выдавшего контролируемую карточку.

controlledVehicleRegistration – VRN и государство-член регистрации транспортного средства, к которому относится контроль.

controlDownloadPeriodBegin и **controlDownloadPeriodEnd** – период, за который впоследствии загружаются данные.

2.52. ControlCardHolderIdentification

Информация, записанная на контрольной карточке, которая относится к идентификации владельца карточки (приложение 1С, требования 360 и 366).

```
ControlCardHolderIdentification ::= SEQUENCE {  
    controlBodyName          Name,  
    controlBodyAddress       Address,  
    cardHolderName           HolderName,  
    cardHolderPreferredLanguage Language  
}
```

controlBodyName – название контрольного органа владельца карточки.

controlBodyAddress – адрес контрольного органа владельца карточки.

cardHolderName – фамилия и имя (имена) владельца контрольной карточки.

cardHolderPreferredLanguage – предпочитаемый язык владельца карточки.

2.53. ControlType

Код, указывающий на действия, проведённые в ходе проверки. Данный тип данных связан с требованиями 126, 274, 299, 327 и 350 в приложении 1С.

```
ControlType ::= OCTET STRING (SIZE(1))
```

Первое поколение:

Присвоение значения – выровненный октет: ‘сvрдxxxx’В (8 бит)

‘с’В	загрузка данных карточки:
‘0’В	во время данной контрольной операции загрузка данных с карточки не произведена,
‘1’В	во время данной контрольной операции загрузка данных с карточки произведена
‘v’В	загрузка данных с БУ:
‘0’В	во время данной контрольной операции загрузка данных с БУ не произведена,
‘1’В	во время данной контрольной операции загрузка данных с БУ произведена
‘р’В	печать:
‘0’В	во время данной контрольной операции печать не произведена,
‘1’В	во время данной контрольной операции печать произведена
‘d’В	отображение:
‘0’В	во время данной контрольной операции данные на дисплей не выводились,
‘1’В	во время данной контрольной операции данные на дисплей выводились
‘xxxx’В	не применяется.

Второе поколение:

Присвоение значения – выровненный октет: ‘c’V (8 бит)

‘c’V загрузка данных карточки:
‘0’V: во время данной контрольной операции загрузка данных с карточки не произведена,
‘1’V: во время данной контрольной операции загрузка данных с карточки произведена

‘v’V загрузка данных с БУ:
‘0’V: во время данной контрольной операции загрузка данных с БУ не произведена,
‘1’V: во время данной контрольной операции загрузка данных с БУ произведена

‘p’V печать:
‘0’V: во время данной контрольной операции печать не произведена,
‘1’V: во время данной контрольной операции печать произведена

‘d’V отображение:
‘0’V: во время данной контрольной операции данные на дисплей не выводились,
‘1’V: во время данной контрольной операции данные на дисплей выводились

‘e’V придорожная проверка калибровки:
‘0’V: во время данной контрольной операции параметры калибровки не проверялись,
‘1’V: во время данной контрольной операции параметры калибровки проверялись

‘xxx’V RFU.

2.54. CurrentDateTime

Текущие дата и время записывающего оборудования.

`CurrentDateTime ::= TimeReal`

Присвоение значения: дополнительно не указывается.

2.55. CurrentDateTimeRecordArray

Второе поколение:

Текущие дата и время и метаданные, используемые в протоколе загрузки.

```
CurrentDateTimeRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize         INTEGER(1..65535),
    noOfRecords        INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF CurrentDateTime
}
```

recordType указывает на тип записи (CurrentDateTime). **Присвоение значения:** см. RecordType

recordSize – размер CurrentDateTime в байтах.

noOfRecords – число записей в массиве.

records – массив записей текущих даты и времени.

2.56. DailyPresenceCounter

Показания счетчика, записанные на карточке водителя или предприятия, которые увеличиваются на единицу за каждый календарный день, в течение которого в БУ была вставлена карточка. Данный тип данных связан с требованиями 266, 299, 320 и 343 в приложении 1С.

`DailyPresenceCounter ::= BCDString(SIZE(2))`

Присвоение значения: порядковый номер с максимальным значением = 9 999, который снова начинается с 0. В момент первой выдачи номера карточки это число устанавливается на 0.

2.57. Datef

Дата, выраженная в числовом формате, которая может сразу выводиться на печать.

```
Datef ::= SEQUENCE {
    year      BCDString(SIZE(2)),
    month     BCDString(SIZE(1)),
    day       BCDString(SIZE(1))
}
```

Присвоение значения:

yyyy	год
mm	месяц
dd	день
'00000000'H	явно выражает отсутствие даты.

2.58. DateOfDayDownloaded

Второе поколение:

Дата и время загрузки.

```
DateOfDayDownloaded ::= TimeReal
```

Присвоение значения: дополнительно не указывается.

2.59. DateOfDayDownloadedRecordArray

Второе поколение:

Дата и время загрузки и метаданные, используемые в протоколе загрузки.

```
DateOfDayDownloadedRecordArray ::= SEQUENCE {
    recordType      RecordType,
    recordSize      INTEGER(1..65535),
    noOfRecords     INTEGER(0..65535),
    records         SET SIZE(noOfRecords) OF
                   DateOfDayDownloaded
}
```

recordType указывает на тип записи (DateOfDayDownloaded). **Присвоение значения:** см. RecordType

recordSize – размер CurrentDateTime в байтах.

noOfRecords – число записей в массиве.

records – массив записей даты и времени загрузки.

2.60. Distance

Пройденное расстояние (результат расчёта разницы между двумя показаниями одометра транспортного средства в километрах).

```
Distance ::= INTEGER(0..216-1)
```

Присвоение значения: Двоичное значение без знака. Значение в км в рабочем диапазоне от 0 до 9 999 км.

2.61. DriverCardApplicationIdentification

Информация, записанная на карточке водителя, которая относится к идентификации приложения карточки (приложение 1С, требования 253 и 278).

Первое поколение:

```
DriverCardApplicationIdentification ::= SEQUENCE {
    typeOfTachographCardId      EquipmentType,
    cardStructureVersion         CardStructureVersion,
    noOfEventsPerType            NoOfEventsPerType,
    noOfFaultsPerType           NoOfFaultsPerType,
    activityStructureLength      CardActivityLengthRange,
    noOfCardVehicleRecords      NoOfCardVehicleRecords,
    noOfCardPlaceRecords       NoOfCardPlaceRecords
}
```

typeOfTachographCardId уточняет реализованный тип карточки.

cardStructureVersion уточняет версию структуры, реализованной на карточке.

noOfEventsPerType – число событий по типу события, которое может быть записано на карточку.

noOfFaultsPerType – число неисправностей по типу неисправности, которое может быть записано на карточку.

activityStructureLength – число байтов, которые могут быть использованы для хранения записей, относящихся к виду деятельности.

noOfCardVehicleRecords – число записей о транспортном средстве, которые могут храниться на карточке.

noOfCardPlaceRecords – число записей о местах, которые могут храниться на карточке.

Второе поколение:

```
DriverCardApplicationIdentification ::= SEQUENCE {
    typeOfTachographCardId      EquipmentType,
    cardStructureVersion         CardStructureVersion,
    noOfEventsPerType            NoOfEventsPerType,
    noOfFaultsPerType           NoOfFaultsPerType,
    activityStructureLength      CardActivityLengthRange,
    noOfCardVehicleRecords      NoOfCardVehicleRecords,
    noOfCardPlaceRecords       NoOfCardPlaceRecords,
    noOfGNSSCDRecords          NoOfGNSSCDRecords,
    noOfSpecificConditionRecords NoOfSpecificConditionRecords
}
```

В дополнение к первому поколению, используются следующие элементы данных:

noOfGNSSCDRecords – число записей о непрерывном времени управления по ГНСС, которые могут храниться на карточке.

noOfSpecificConditionRecords – число записей об особых условиях, которые могут храниться на карточке.

2.62. DriverCardHolderIdentification

Информация, записанная на карточке водителя, которая относится к идентификации владельца карточки (приложение 1С, требования 256 и 281).

```
DriverCardHolderIdentification ::= SEQUENCE {
    cardHolderName           HolderName,
    cardHolderBirthDate      Datef,
    cardHolderPreferredLanguage Language
}
```

cardHolderName – фамилия и имя (имена) владельца карточки водителя.

cardHolderBirthDate – дата рождения владельца карточки водителя.

cardHolderPreferredLanguage – предпочитаемый язык владельца карточки.

2.63. DSRCSecurityData

Второе поколение:

Простая текстовая информация и MAC, передаваемые через DSRC из тахографа в устройство дистанционного опроса (RI); более подробно см. приложение 11, часть Б, глава 13.

```
DSRCSecurityData ::= SEQUENCE {
    tagLengthPlainText      OCTET STRING(SIZE(2)),
    currentDate             CurrentDateTime,
    counter                 INTEGER(0..224-1),
    vuSerialNumber          VuSerialNumber,
    dsRCMKVersionNumber     INTEGER(SIZE(1)),
    tagLengthMac            OCTET STRING(SIZE(2)),
    mac                    MAC
}
```

tagLength – часть кодирования DER-TLV, которая устанавливается как '81 10' (см. приложение 11, часть Б, глава 13).

currentDateTime – текущие дата и время бортового устройства.

counter перечисляет сообщения RTM.

vuSerialNumber – серийный номер бортового устройства.

dsRCMKVersionNumber – номер версии главного ключа DSRC, из которого получены ключи DSRC, конкретно связанные с БУ.

tagLengthMac – маркировка и длина объекта данных MAC как части кодирования DER-TLV. Маркировка устанавливается как '8E', длина кодирует октетную длину MAC (см. приложение 11, часть Б, глава 13).

mac – это значение MAC, вычисленное при помощи сообщения RTM (см. приложение 11, часть Б, глава 13).

2.64. EGFCertificate

Второе поколение:

Сертификат открытого ключа внешнего устройства ГНСС для взаимной проверки подлинности с БУ. Структура данного сертификата представлена в приложении 11.

```
EGFCertificate ::= Certificate
```

2.65. EmbedderIcAssemblerId

Информация об установщике ИС.

```
EmbedderIcAssemblerId ::= SEQUENCE{
```

```

countryCode          IA5String(SIZE(2)),
moduleEmbedder      BCDString(SIZE(2)),
manufacturerInformation OCTET STRING(SIZE(1))
}

```

countryCode – двухбуквенный код страны установщика модуля в соответствии с ISO 3166.

moduleEmbedder – идентификатор установщика модуля.

manufacturerInformation – для внутреннего пользования производителя.

2.66. EntryTypeDailyWorkPeriod

Код, позволяющий провести различие между началом и концом ввода данных о месте дневного периода работы и условиями ввода.

Первое поколение

```

EntryTypeDailyWorkPeriod ::= INTEGER {
  Begin, related time = card insertion time or time of entry      (0),
  End,   related time = card withdrawal time or time of entry    (1),
  Begin, related time manually entered (start time)              (2),
  End,   related time manually entered (end of work period)      (3),
  Begin, related time assumed by VU                               (4),
  End,   related time assumed by VU                               (5)
}

```

Присвоение значения: в соответствии с ISO/IEC8824-1.

Второе поколение

```

EntryTypeDailyWorkPeriod ::= INTEGER {
  Begin, related time = card insertion time or time of entry      (0),
  End,   related time = card withdrawal time or time of entry    (1),
  Begin, related time manually entered (start time)              (2),
  End,   related time manually entered (end of work period)      (3),
  Begin, related time assumed by VU                               (4),
  End,   related time assumed by VU                               (5),
  Begin, related time based on GNSS data                          (6),
  End,   related time based on GNSS data                          (7)
}

```

Присвоение значения: в соответствии с ISO/IEC8824-1.

2.67. EquipmentType

Код, позволяющий провести различие между различными типами оборудования в связи с использованием тахографа.

```
EquipmentType ::= INTEGER(0..255)
```

Первое поколение:

```

--Reserved          (0),
--Driver Card      (1),
--Workshop Card    (2),
--Control Card     (3),
--Company Card     (4),
--Manufacturing Card (5),
--Vehicle Unit     (6),
--Motion Sensor    (7),
--RFU              (8..255)

```

Присвоение значения: В соответствии со стандартом ISO/IEC8824-1.

Значение 0 зарезервировано для целей указания государства-члена или Европы в поле данных США сертификатов.

Второе поколение:

Используются такие же значения, как и для первого поколения, со следующими дополнениями:

--GNSS Facility	(8),
--Remote Communication Module	(9),
--ITS interface module	(10),
--Plaque	(11), -- may be used in SealRecord
--M1/N1 Adapter	(12), -- may be used in SealRecord
--European Root CA (ERCA)	(13),
--Member State CA (MSCA)	(14),
--External GNSS connection	(15), -- may be used in SealRecord
--Unused	(16), -- used in SealDataVu
--RFU	(17..255)

Примечание: Значения второго поколения в отношении таблички, адаптера и внешнего соединения ГНСС, а также значения первого поколения в отношении бортового устройства и датчика движения могут использоваться в SealRecord, т.е. если применимы.

2.68. EuropeanPublicKey

Первое поколение:

Европейский открытый ключ.

EuropeanPublicKey ::= PublicKey

2.69. EventFaultRecordPurpose

Код, указывающий на причину регистрации события или неисправности.

EventFaultRecordPurpose ::= OCTET STRING (SIZE(1))

Присвоение значения:

'00' H	одно из 10 недавних (последних) событий или неисправностей
'01' H	самое продолжительное событие за один из последних 10 дней данного случая
'02' H	одно из 5 самых продолжительных событий за последние 365 дней
'03' H	последнее событие за один из последних 10 дней данного случая
'04' H	самое серьезное событие за один из последних 10 дней данного случая
'05' H	одно из 5 самых серьезных событий за последние 365 дней
'06' H	первое событие или неисправность, наступившие после последней калибровки
'07' H	текущее/продолжающееся событие или неисправность
'08' H - '7F' H	RFU
'80' H - 'FF' H	относится конкретно к производителю

2.70. EventFaultType

Код, отображающий событие или неисправность.

EventFaultType ::= OCTET STRING (SIZE(1))

Присвоение значения:

Первое поколение:

\0x' H	общие события,
\00' H	дополнительно не уточняется,
\01' H	ввод недействительной карточки,
\02' H	несовместимость карточек,
\03' H	совпадение во времени,
\04' H	управление без соответствующей карточки,
\05' H	ввод карточки во время управления,
\06' H	неправильное завершение последнего сеанса использования карточки,
\07' H	превышение скорости,
\08' H	прекращение электропитания,
\09' H	ошибочные данные о движении,
\0A' H	противоречивые данные о движении транспортного средства,
\0B' - \0F' H	RFU,
\1x' H	события, связанные с попыткой нарушения защиты бортового устройства,
\10' H	дополнительно не уточняется,
\11' H	сбой в аутентификации датчика движения,
\12' H	сбой в аутентификации карточки тахографа,
\13' H	несанкционированное изменение датчика движения,
\14' H	ошибка при проверке целостности введённых данных карточки
\15' H	ошибка при проверке целостности сохранённых пользовательских данных,
\16' H	ошибка при внутренней передаче данных,
\17' H	несанкционированное открытие корпуса,
\18' H	саботаж аппаратного обеспечения,
\19' H - \1F' H	RFU,
\2x' H	события, связанные с попыткой нарушения защиты датчика,
\20' H	дополнительно не уточняется,
\21' H	сбой в аутентификации,
\22' H	ошибка при проверке целостности сохранённых данных,
\23' H	ошибка при внутренней передаче данных,
\24' H	несанкционированное открытие корпуса,
\25' H	саботаж аппаратного обеспечения,
\26' H - \2F' H	RFU,
\3x' H	неисправность записывающего оборудования,
\30' H	дополнительно не уточняется,
\31' H	внутренние неполадки в БУ,
\32' H	неисправность принтера,
\33' H	неисправность дисплея,
\34' H	неисправность при загрузке,
\35' H	неисправность датчика,
\36' H - \3F' H	RFU,
\4x' H	неисправности карточки,
\40' H	дополнительно не уточняется,
\41' H - \4F' H	RFU,
\50' H - \7F' H	RFU,
\80' H - \FF' H	относится конкретно к производителю.

Второе поколение:

Используются такие же значения, как и для первого поколения, со следующими дополнениями:

\0В'Н	нестыковка во времени (ГНСС и внутренние часы БУ),
\0С' - \0F'Н	RFU,
\5х'Н	неисправности, связанные с ГНСС,
\50'Н	дополнительно не уточняется,
\51'Н	внутренняя неисправность приёмника ГНСС,
\52'Н	внешняя неисправность приёмника ГНСС,
\53'Н	неисправность внешней связи ГНСС,
\54'Н	отсутствие данных ГНСС о местоположении,
\55'Н	обнаружение взлома ГНСС,
\56'Н	истёкший срок действия сертификата внешнего устройства ГНСС,
\57'Н - \5F'Н	RFU,
\6х'Н	неисправности, связанные с модулем удалённой связи,
\60'Н	дополнительно не уточняется,
\61'Н	неисправность модуля удалённой связи,
\62'Н	неисправность связи модуля удалённой связи,
\63'Н - \6F'Н	RFU,
\7х'Н	неисправности интерфейса ИТС,
\70'Н	дополнительно не уточняется,
\71'Н - \7F'Н	RFU.

2.71. ExtendedSealIdentifier

Второе поколение:

Расширенный идентификатор пломбы для уникальной идентификации пломбы (приложение 1С, требование 401).

```

ExtendedSealIdentifier ::= SEQUENCE{
    manufacturerCode      OCTET STRING (SIZE(2)),
    sealIdentifier         OCTET STRING (SIZE(6))
}

```

manufacturerCode – код производителя пломбы.

sealIdentifier – идентификатор пломбы, уникальный для производителя.

2.72. ExtendedSerialNumber

Уникальная идентификация оборудования. Она может использоваться в качестве идентификатора открытого ключа оборудования.

Первое поколение:

```

ExtendedSerialNumber ::= SEQUENCE{
    serialNumber          INTEGER(0..232-1),
    monthYear             BCDString(SIZE(2)),
    type                  OCTET STRING(SIZE(1)),
    manufacturerCode      ManufacturerCode
}

```

serialNumber – серийный номер оборудования, уникальный для данного производителя, типа оборудования и месяца и года, указанных ниже.

monthYear – идентификация месяца и года изготовления (или присвоение серийного номера).

Присвоение значения: код BCD месяца (две цифры) и года (две последние цифры).

type – идентификатор типа оборудования.

Присвоение значения: связано с конкретным производителем, с зарезервированным значением 'FFh'.

manufacturerCode – цифровой код идентификации производителя оборудования утверждённого типа.

Второе поколение:

```

ExtendedSerialNumber ::= SEQUENCE{
    serialNumber          INTEGER(0..232-1),
    monthYear             BCDString(SIZE(2)),
    type                  EquipmentType,
    manufacturerCode      ManufacturerCode
}

```

serialNumber – см. первое поколение

monthYear – см. первое поколение

type указывает тип оборудования

manufacturerCode – см. первое поколение.

2.73. FullCardNumber

Код, полностью идентифицирующий карточку тахографа.

```

FullCardNumber ::= SEQUENCE {
    cardType              EquipmentType,
    cardIssuingMemberState NationNumeric,
    cardNumber            CardNumber
}

```

cardType – тип карточки тахографа.

cardIssuingMemberState – код государства-члена, выдавшего карточку.

cardNumber – номер карточки.

2.74. FullCardNumberAndGeneration

Второе поколение:

Код, полностью идентифицирующий карточку тахографа и её поколение.

```
FullCardNumberAndGeneration ::= SEQUENCE {  
    fullCardNumber          FullCardNumber,  
    generation              Generation  
}
```

fullcardNumber идентифицирует карточку тахографа.

generation указывает на поколение используемой карточки тахографа.

2.75. Generation

Второе поколение:

Указывает на поколение используемого тахографа.

```
Generation ::= INTEGER(0..255)
```

Присвоение значения:

'00'H	RFU
'01'H	первое поколение
'02'H	второе поколение
'03'H..'FF'H	RFU

2.76. GeoCoordinates

Второе поколение:

Геокоординаты кодируются как целые числа. Такие целые числа представляют собой показатель кодирования $\pm DDMM.M$ по широте и $\pm DDDMM.M$ по долготе. В данном случае $\pm DD$ и соответственно $\pm DDD$ обозначают градусы и минуты $MM.M$.

```
GeoCoordinates ::= SEQUENCE {  
    latitude                INTEGER(-90000..90001),  
    longitude               INTEGER(-180000..180001)  
}
```

latitude кодируется как показатель (коэффициент 10) выражения $\pm DDMM.M$.

longitude кодируется как показатель (коэффициент 10) выражения $\pm DDDMM.M$.

2.77. GNSSAccuracy

Второе поколение:

Точность данных ГНСС о местоположении (определение ее). Эта точность кодируется как целое число и является показателем (коэффициент 10) значения X.Y из строки GSA NMEA.

```
GNSSAccuracy ::= INTEGER(1..100)
```

2.78. GNSSContinuousDriving

Второе поколение:

Информация, записанная на карточке водителя или мастерской, которая относится к местоположению транспортного средства по ГНСС, если непрерывное время управления водителем достигает значения, кратного трём часам (приложение 1С, требования 306 и 354).

```
GNSSContinuousDriving := SEQUENCE {  
    gnssCDPointerNewestRecord      INTEGER(0..NoOfGNSSCDRecords -1),  
    gnssContinuousDrivingRecords  SET SIZE(NoOfGNSSCDRecords) OF  
                                   GNSSContinuousDrivingRecord  
}
```

gnssCDPointerNewestRecord – индекс последней обновлённой записи ГНСС, касающейся непрерывного управления.

Присвоение значения: Число, соответствующее числовому показателю записи данных ГНСС о непрерывном управлении, которое начинается с 0 в случае первой регистрации регистрации записей ГНСС, касающихся непрерывного управления, в структуре.

gnssContinuousDrivingRecords – массив записей, содержащих дату и время, когда непрерывное управление достигает значения, кратного трём часам, и информацию о местоположении транспортного средства.

2.79. GNSSContinuousDrivingRecord

Второе поколение:

Информация, записанная на карточке водителя или мастерской, которая относится к местоположению транспортного средства по ГНСС, если непрерывное время управления водителем достигает значения, кратного трём часам (приложение 1С, требования 305 и 353).


```
GNSSContinuousDrivingRecord ::= SEQUENCE {
    timeStamp                TimeReal,
    gnssPlaceRecord         GNSSPlaceRecord
}
```

timeStamp – дата и время, когда непрерывное время управления владельцем карточки достигает значения, кратного трём часам.

gnssPlaceRecord содержит информацию, связанную с местоположением транспортного средства.

2.80. GNSSPlaceRecord

Второе поколение:

Информация, которая имеет отношение к местоположению транспортного средства по ГНСС (приложение 1С, требования 108, 109, 110, 296, 305, 347 и 353).

```
GNSSPlaceRecord ::= SEQUENCE {
    timeStamp                TimeReal,
    gnssAccuracy            GNSSAccuracy,
    geoCoordinates         GeoCoordinates
}
```

timeStamp – дата и время, когда установлено местоположение транспортного средства по ГНСС.

gnssAccuracy – точность данных ГНСС о местоположении.

geoCoordinates – местоположение, зарегистрированное с помощью ГНСС.

2.81. HighResOdometer

Показания одометра транспортного средства. общее расстояние, пройденное транспортным средством за период его эксплуатации.

```
HighResOdometer ::= INTEGER(0..232-1)
```

Присвоение значения: Двоичное значение без знака. Значение в 1/200 км в рабочем диапазоне от 0 до 21 055 406 км.

2.82. HighResTripDistance

Расстояние, пройденное за весь или часть рейса.

```
HighResTripDistance ::= INTEGER(0..232-1)
```

Присвоение значения: Двоичное значение без знака. Значение в 1/200 км в рабочем диапазоне от 0 до 21 055 406 км.

2.83. HolderName

Фамилия и имя (имена) владельца карточки.

```
HolderName ::= SEQUENCE {
    holderSurname            Name,
    holderFirstNames       Name
}
```

holderSurname – фамилия владельца. Фамилия не включает в себя обращение.

Присвоение значения: если карточка не именная, позиция holderSurname содержит ту же информацию, что и companyname или workshopName, или controlBodyName.

holderFirstNames – имя (имена) и инициалы владельца.

2.84. InternalGNSSReceiver

Второе поколение:

Информация о том, является ли приёмник ГНСС внешним или внутренним устройством бортового устройства. True – приёмник ГНСС встроен в БУ. False – приёмник ГНСС является внешним устройством.

```
InternalGNSSReceiver ::= BOOLEAN
```

2.85. K-ConstantOfRecordingEquipment

Постоянная величина записывающего оборудования (определение m).

```
K-ConstantOfRecordingEquipment ::= INTEGER(0..216-1)
```

Присвоение значения: импульсы на километр в рабочем диапазоне от 0 до 64 255 имп./км.

2.86. KeyIdentifier

Уникальный идентификатор открытого ключа, используемого для назначения и выбора ключа. Также идентифицирует владельца ключа.

```
KeyIdentifier ::= CHOICE {  
    extendedSerialNumber          ExtendedSerialNumber,  
    certificateRequestID          CertificateRequestID,  
    certificationAuthorityKID     CertificationAuthorityKID  
}
```

Первый вариант выбора позволяет назначить открытый ключ бортового устройства или карточки тахографа.

Второй вариант выбора позволяет назначить открытый ключ бортового устройства (если в момент создания сертификата серийный номер бортового устройства неизвестен).

Третий вариант выбора позволяет назначить открытый ключ государства-члена.

2.87. KMWCKey

Второе поколение:

Ключ AES и связанная с ним версия ключа используются для соединения БУ и датчика движения. Более подробно см. приложение 11.

```
KMWCKey ::= SEQUENCE {  
    kMWCKey          AESKey,  
    keyVersion       INTEGER (SIZE(1))  
}
```

kMWCKey – длина ключа AES, соединённого с ключом, используемым для соединения БУ и датчика движения.

keyVersion обозначает основную версию ключа AES.

2.88. Language

Код, идентифицирующий язык.

```
Language ::= IA5String(SIZE(2))
```

Присвоение значения: код в виде двух строчных букв в соответствии со стандартом ISO 639.

2.89. LastCardDownload

Дата и время, записанные на карточке водителя, последней загрузки данных с карточки (для иных целей, кроме контроля) (приложение 1С, требования 257 и 282). Эта дата может обновляться БУ или любым считывающим устройством.

```
LastCardDownload ::= TimeReal
```

Присвоение значения: дополнительно не указывается.

2.90. LinkCertificate

Второе поколение:

Сертификат связи между парами ключей European Root CA.

LinkCertificate ::= Certificate

2.91. L-TyreCircumference

Фактическая окружность шин колёс (определение u).

L-TyreCircumference ::= INTEGER(0.. $2^{16}-1$)

Присвоение значения: двоичное значение без знака; значение в 1/8 мм в рабочем диапазоне от 0 до 8 031 мм.

2.92. MAC

Второе поколение:

криптографическая контрольная сумма длиной 8, 12 или 16 байтов, соответствующая наборам шифров, представленным в приложении 11.

```
MAC ::= CHOICE {  
    mac8                OCTET STRING (SIZE(8)),  
    mac12               OCTET STRING (SIZE(12)),  
    mac16               OCTET STRING (SIZE(16))  
}
```

2.93. ManualInputFlag

Код, позволяющий определить, ввёл ли владелец карточки данные о деятельности водителя вручную в момент ввода карточки или нет (приложение 1B, требование 081 и приложение 1C, требование 102).

```
ManualInputFlag ::= INTEGER {  
    noEntry              (0)  
    manualEntries       (1)  
}
```

Присвоение значения: дополнительно не указывается.

2.94. ManufacturerCode

Код идентификации производителя оборудования утверждённого типа.

ManufacturerCode ::= INTEGER(0..255)

Лаборатория, уполномоченная проводить испытания эксплуатационной совместимости, составляет и публикует перечень кодов производителей на своём веб-сайте (приложение 1C, требование 454).

ManufacturerCodes временно присваиваются разработчикам оборудования тахографов при подаче заявки лаборатории, уполномоченной проводить испытания эксплуатационной совместимости.

2.95. ManufacturerSpecificEventFaultData

Второе поколение:

Коды ошибок, связанные с конкретными производителями, упрощают анализ ошибок и обслуживание бортовых устройств.

```
ManufacturerSpecificEventFaultData ::= SEQUENCE {
    manufacturerCode      ManufacturerCode,
    manufacturerSpecificErrorCode OCTET STRING(SIZE(3))
}
```

manufacturerCode идентифицирует производителя бортового устройства.

manufacturerSpecificErrorCode – код ошибки, связанный с конкретным производителем.

2.96. MemberStateCertificate

Сертификат открытого ключа государства-члена, выданный европейским сертификационным органом.

```
MemberStateCertificate ::= Certificate
```

2.97. MemberStateCertificateRecordArray

Второе поколение:

Сертификат государства-члена и метаданные, используемые в протоколе загрузки.

```
MemberStateCertificateRecordArray ::= SEQUENCE {
    recordType      RecordType,
    recordSize      INTEGER(1..65535),
    noOfRecords     INTEGER(0..65535),
    records         SET SIZE(noOfRecords) OF
                  MemberStateCertificate
}
```

recordType указывает на тип записи (MemberStateCertificate). **Присвоение значения:** см. RecordType

recordSize – размер MemberStateCertificate в байтах.

noOfRecords – число записей в массиве. Значение устанавливается на 1, так как сертификаты могут быть разной длины.

records – набор сертификатов государств-членов.

2.98. MemberStatePublicKey

Первое поколение:

Открытый ключ государства-члена.

```
MemberStatePublicKey ::= PublicKey
```

2.99. Name

Название.

```
Name ::= SEQUENCE {
    codePage      INTEGER (0..255),
    name          OCTET STRING (SIZE(35))
}
```

codePage определяет набор символов, указанный в главе 4,

name – название, закодированное с помощью указанного набора символов.

2.100. NationAlpha

Буквенное обозначение страны соответствует отличительным знакам, используемым на транспортных средствах в международных перевозках (Венская конвенция ООН о дорожном движении 1968 г.).

`NationAlpha ::= IA5String(SIZE(3))`

NationAlpha и цифровые коды включаются в перечень, составляемый на веб-сайте лаборатории, назначенной для проведения испытаний эксплуатационной совместимости, как указано в требовании 440 приложения 1С.

2.101. NationNumeric

Цифровое обозначение страны.

`NationNumeric ::= INTEGER(0 .. 255)`

Присвоение значения: см. тип данных 2.100 (NationAlpha).

Любое изменение или обновление спецификации буквенного или цифрового обозначения страны, описанной в предыдущем пункте, осуществляется только после того, как назначенная лаборатория получит отзывы производителей бортовых устройств цифровых и «умных» тахографов утверждённого типа.

2.102. NoOfCalibrationRecords

Число записей калибровки, которое может храниться на карточке.

Первое поколение:

`NoOfCalibrationRecords ::= INTEGER(0..255)`

Присвоение значения: см. приложение 2.

Второе поколение:

`NoOfCalibrationRecords ::= INTEGER(0..216-1)`

Присвоение значения: см. приложение 2.

2.103. NoOfCalibrationsSinceDownload

Счётчик, указывающий на число калибровок, произведённых с карточкой мастерской после последней загрузки данных с этой карточки (приложение 1С, требования 317 и 340).

`NoOfCalibrationsSinceDownload ::= INTEGER(0..216-1)`

Присвоение значения: дополнительно не указывается.

2.104. NoOfCardPlaceRecords

Число записей о месте, которое может храниться на карточке водителя или мастерской.

Первое поколение:

`NoOfCardPlaceRecords ::= INTEGER(0..255)`

Присвоение значения: см. приложение 2.

Второе поколение:

`NoOfCardPlaceRecords ::= INTEGER(0..216-1)`

Присвоение значения: см. приложение 2.

2.105. NoOfCardVehicleRecords

Число записей об используемых транспортных средствах, которое может храниться на карточке водителя или мастерской.

`NoOfCardVehicleRecords ::= INTEGER(0.. 216-1)`

Присвоение значения: см. приложение 2.

2.106. NoOfCardVehicleUnitRecords

Второе поколение:

Число записей об используемых бортовых устройствах, которое может храниться на карточке водителя или мастерской.

NoOfCardVehicleUnitRecords ::= INTEGER(0.. 2¹⁶-1)

Присвоение значения: см. приложение 2.

2.107. NoOfCompanyActivityRecords

Число записей о действиях предприятия, которое может храниться на карточке предприятия.

NoOfCompanyActivityRecords ::= INTEGER(0.. 2¹⁶-1)

Присвоение значения: см. приложение 2.

2.108. NoOfControlActivityRecords

Число записей о контрольных действиях, которое может храниться на контрольной карточке.

NoOfControlActivityRecords ::= INTEGER(0.. 2¹⁶-1)

Присвоение значения: см. приложение 2.

2.109. NoOfEventsPerType

Число событий по типу события, которое может быть записано на карточку.

NoOfEventsPerType ::= INTEGER(0..255)

Присвоение значения: см. приложение 2.

2.110. NoOfFaultsPerType

Число неисправностей по типу неисправности, которое может быть записано на карточку.

NoOfFaultsPerType ::= INTEGER(0..255)

Присвоение значения: см. приложение 2.

2.111. NoOfGNSSCDRecords

Второе поколение:

Число записей о непрерывном времени управления по ГНСС, которые могут храниться на карточке.

NoOfGNSSCDRecords ::= INTEGER(0..2¹⁶-1)

Присвоение значения: см. приложение 2.

2.112. NoOfSpecificConditionRecords

Второе поколение:

Число записей об особых условиях, которые могут храниться на карточке.

NoOfSpecificConditionRecords ::= INTEGER(0..2¹⁶-1)

Присвоение значения: см. приложение 2.

2.113. OdometerShort

Показания одометра транспортного средства в краткой форме.

OdometerShort ::= INTEGER(0..2²⁴-1)

Присвоение значения: Двоичное значение без знака. Значение в км в рабочем диапазоне от 0 до 9 999 999 км.

2.114. OdometerValueMidnight

Показания одометра транспортного средства в полночь на данный день (приложение 1В, требование 090 и приложение 1С, требование 113).

```
OdometerValueMidnight ::= OdometerShort
```

Присвоение значения: дополнительно не указывается.

2.115. OdometerValueMidnightRecordArray

Второе поколение:

OdometerValueMidnight и метаданные, используемые в протоколе загрузки.

```
OdometerValueMidnightRecordArray ::= SEQUENCE {
    recordType           RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF
                       OdometerValueMidnight
}
```

recordType указывает на тип записи (OdometerValueMidnight). **Присвоение значения:** см. RecordType

recordSize – размер OdometerValueMidnight в байтах.

noOfRecords – число записей в массиве.

records – массив записей OdometerValueMidnight.

2.116. OverspeedNumber

Число случаев превышения скорости после последнего контроля за превышением скорости.

```
OverspeedNumber ::= INTEGER(0..255)
```

Присвоение значения: 0 означает, что после последнего контроля за превышением скорости случаев превышения скорости не было, 1 означает, что после последнего контроля за превышением скорости был один случай превышения скорости... 255 означает, что после последнего контроля за превышением скорости было 255 или больше случаев превышения скорости.

2.117. PlaceRecord

Информация, касающаяся места, в котором начинается или заканчивается дневной период работы (приложение 1С, требования 108, 271, 296, 324 и 347).

Первое поколение:

```
PlaceRecord ::= SEQUENCE {
    entryTime           TimeReal,
    entryTypeDailyWorkPeriod EntryTypeDailyWorkPeriod,
    dailyWorkPeriodCountry NationNumeric,
    dailyWorkPeriodRegion RegionNumeric,
    vehicleOdometerValue OdometerShort
}
```

entryTime – дата и время, связанные с вводом данных.

entryTypeDailyWorkPeriod – тип ввода.

dailyWorkPeriodCountry – введённая страна.

dailyWorkPeriodRegion – введённый регион.

vehicleOdometerValue – показания одометра в момент ввода данных о месте.

Второе поколение:

```

PlaceRecord ::= SEQUENCE {
    entryTime                TimeReal,
    entryTypeDailyWorkPeriod EntryTypeDailyWorkPeriod,
    dailyWorkPeriodCountry   NationNumeric,
    dailyWorkPeriodRegion    RegionNumeric,
    vehicleOdometerValue     OdometerShort,
    entryGNSSPlaceRecord     GNSSPlaceRecord
}

```

В дополнение к первому поколению, используется следующий компонент:

entryGNSSPlaceRecord – зафиксированные место и время.

2.118. PreviousVehicleInfo

Информация, касающаяся транспортного средства, использованного водителем ранее, в момент ввода его карточки в бортовое устройство (приложение 1B, требование 081 и приложение 1C, требование 102).

Первое поколение:

```

PreviousVehicleInfo ::= SEQUENCE {
    vehicleRegistrationIdentification VehicleRegistrationIdentification,
    cardWithdrawalTime              TimeReal
}

```

vehicleRegistrationIdentification – VRN и государство-член регистрации транспортного средства.

cardWithdrawalTime – дата и время извлечения карточки.

Второе поколение:

```

PreviousVehicleInfo ::= SEQUENCE {
    vehicleRegistrationIdentification VehicleRegistrationIdentification,
    cardWithdrawalTime              TimeReal,
    vuGeneration                    Generation
}

```

В дополнение к первому поколению, используется следующий элемент данных:

vuGeneration идентифицирует поколение бортового устройства.

2.119. PublicKey

Первое поколение:

Открытый ключ RSA.

```

PublicKey ::= SEQUENCE {
    rsaKeyModulus          RSAKeyModulus,
    rsaKeyPublicExponent   RSAKeyPublicExponent
}

```

rsaKeyModulus – модуль пары ключей.

rsaKeyPublicExponent – открытый показатель пары ключей.

2.120. RecordType

Второе поколение:

Указание на тип записи. Такой тип данных используется в RecordArrays.

`RecordType ::= OCTET STRING(SIZE(1))`

Присвоение значения:

'01' H	ActivityChangeInfo,
'02' H	CardSlotsStatus,
'03' H	CurrentDateTime,
'04' H	MemberStateCertificate,
'05' H	OdometerValueMidnight,
'06' H	DateOfDayDownloaded,
'07' H	SensorPaired,
'08' H	Signature,
'09' H	SpecificConditionRecord,
'0A' H	VehicleIdentificationNumber,
'0B' H	VehicleRegistrationNumber,
'0C' H	VuCalibrationRecord,
'0D' H	VuCardIWRecord,
'0E' H	VuCardRecord,
'0F' H	VuCertificate,
'10' H	VuCompanyLocksRecord,
'11' H	VuControlActivityRecord,
'12' H	VuDetailedSpeedBlock,
'13' H	VuDownloadablePeriod,
'14' H	VuDownloadActivityData,
'15' H	VuEventRecord,
'16' H	VuGNSSCDRecord,
'17' H	VuITSConsentRecord,
'18' H	VuFaultRecord,
'19' H	VuIdentification,
'1A' H	VuOverSpeedingControlData,
'1B' H	VuOverSpeedingEventRecord,
'1C' H	VuPlaceDailyWorkPeriodRecord,
'1D' H	VuTimeAdjustmentGNSSRecord,
'1E' H	VuTimeAdjustmentRecord,
'1F' H	VuPowerSupplyInterruptionRecord,
'20' H	SensorPairedRecord,
'21' H	SensorExternalGNSSCoupledRecord,
'22' H - '7F' H	RFU,
'80' H - 'FF' H	относится к конкретному производителю.

2.121. RegionAlpha

Буквенное обозначение региона в конкретной стране.

`RegionAlpha ::= IA5STRING(SIZE(3))`

Первое поколение:

Присвоение значения:

' '	Нет информации,
Испания:	
'AN'	Andalucía,
'AR'	Aragón,
'AST'	Asturias,
'C'	Cantabria,
'CAT'	Cataluña,

'CL'	Castilla-León,
'CM'	Castilla-La-Mancha,
'CV'	Valencia,
'EXT'	Extremadura,
'G'	Galicia,
'IB'	Baleares,
'IC'	Canarias,
'LR'	La Rioja,
'M'	Madrid,
'MU'	Murcia,
'NA'	Navarra,
'PV'	País Vasco

Второе поколение:

Коды RegionAlpha включаются в перечень, составляемый на веб-сайте лаборатории, назначенной для проведения испытаний эксплуатационной совместимости.

2.122. RegionNumeric

Цифровое обозначение региона в конкретной стране.

RegionNumeric ::= OCTET STRING (SIZE(1))

Первое поколение:

Присвоение значения:

'00'N	Нет информации,
Испания:	
'01'N	Andalucía,
'02'N	Aragón,
'03'N	Asturias,
'04'N	Cantabria,
'05'N	Cataluña,
'06'N	Castilla-León,
'07'N	Castilla-La-Mancha,
'08'N	Valencia,
'09'N	Extremadura,
'0A'N	Galicia,
'0B'N	Baleares,
'0C'N	Canarias,
'0D'N	La Rioja,
'0E'N	Madrid,
'0F'N	Murcia,
'10'N	Navarra,
'11'N	País Vasco

Второе поколение:

Коды RegionNumeric включаются в перечень, составляемый на веб-сайте лаборатории, назначенной для проведения испытаний эксплуатационной совместимости.

2.123. RemoteCommunicationModuleSerialNumber

Второе поколение:

Серийный номер модуля удалённой связи.

```
RemoteCommunicationModuleSerialNumber ::= ExtendedSerialNumber
```

2.124. RSAKeyModulus

Первое поколение:

Модуль пары ключей RSA.

```
RSAKeyModulus ::= OCTET STRING (SIZE(128))
```

Присвоение значения: Не указано.

2.125. RSAKeyPrivateExponent

Первое поколение:

Закрытый показатель пары ключей RSA.

```
RSAKeyPrivateExponent ::= OCTET STRING (SIZE(128))
```

Присвоение значения: Не указано.

2.126. RSAKeyPublicExponent

Первое поколение:

Открытый показатель пары ключей RSA.

```
RSAKeyPublicExponent ::= OCTET STRING (SIZE(8))
```

Присвоение значения: Не указано.

2.127. RtmData

Второе поколение:

Определение данного типа данных см. в приложении 14.

2.128. SealDataCard

Второе поколение:

Данный тип данных хранит информацию о пломбах, наложенных на различные компоненты транспортного средства, и предназначен для хранения на карточке. Данный тип данных связан с требованием 337 приложения 1С.

```
SealDataCard ::= SEQUENCE {  
    noOfSealRecords          INTEGER(1..5),  
    sealRecords              SET SIZE(noOfSealRecords) OF SealRecord  
}
```

noOfSealRecords – число записей в sealRecords.

sealRecords – массив записей о пломбах.

2.129. SealDataVu

Второе поколение:

Данный тип данных хранит информацию о пломбах, наложенных на различные компоненты транспортного средства, и предназначен для хранения в бортовом устройстве.

```
SealDataVu ::= SEQUENCE SIZE(5) OF {
    sealRecords          SealRecord
}
```

sealRecords – массив записей о пломбах. Если число пломб менее 5, значение EquipmentType во всех неиспользуемых sealRecords устанавливается на 16, т.е. не используется.

2.130. SealRecord

Второе поколение:

Данный тип данных хранит информацию о пломбе, наложенной на компонент. Данный тип данных связан с требованием 337 приложения 1С.

```
SealRecord ::= SEQUENCE {
    equipmentType          EquipmentType,
    extendedSealIdentifier ExtendedSealIdentifier
}
```

equipmentType идентифицирует тип оборудования, на которое наложена пломба.

extendedSealIdentifier – идентификатор пломбы, наложенной на оборудование.

2.131. SensorApprovalNumber

Номер официального утверждения типа датчика.

Первое поколение:

```
SensorApprovalNumber ::= IA5String(SIZE(8))
```

Присвоение значения: Не указано.

Второе поколение:

```
SensorApprovalNumber ::= IA5String(SIZE(16))
```

Присвоение значения:

Номер официального утверждения указывается в соответствии с публикацией на соответствующем веб-сайте Европейской комиссии, например, с дефисами, если они есть. Номер официального утверждения смещён влево.

2.132. SensorExternalGNSSApprovalNumber

Второе поколение:

Номер официального утверждения типа внешнего устройства ГНСС.

```
SensorExternalGNSSApprovalNumber ::= IA5String(SIZE(16))
```

Присвоение значения:

Номер официального утверждения указывается в соответствии с публикацией на соответствующем веб-сайте Европейской комиссии, например, с дефисами, если они есть. Номер официального утверждения смещён влево.

2.133. SensorExternalGNSSCoupledRecord

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к идентификации внешнего устройства ГНСС, соединённого с бортовым устройством (приложение 1С, требование 100).

```
SensorExternalGNSSCoupledRecord ::= SEQUENCE {  
    sensorSerialNumber          SensorGNSSSerialNumber,  
    sensorApprovalNumber       SensorExternalGNSSApprovalNumber,  
    sensorCouplingDate         SensorGNSSCouplingDate  
}
```

sensorSerialNumber – серийный номер внешнего устройства ГНСС, соединённого с бортовым устройством.

sensorApprovalNumber – номер официального утверждения внешнего устройства ГНСС.

sensorCouplingDate – дата соединения внешнего устройства ГНСС с бортовым устройством.

2.134. SensorExternalGNSSIdentification

Второе поколение:

Информация, связанная с идентификацией внешнего устройства ГНСС (приложение 1С, требование 98).

```
SensorExternalGNSSIdentification ::= SEQUENCE {  
    sensorSerialNumber          SensorGNSSSerialNumber,  
    sensorApprovalNumber       SensorExternalGNSSApprovalNumber,  
    sensorSCIdentifier          SensorExternalGNSSSCIdentifier,  
    sensorOSIdentifier         SensorExternalGNSSOSIdentifier  
}
```

sensorSerialNumber – расширенный серийный номер внешнего устройства ГНСС.

sensorApprovalNumber – номер официального утверждения внешнего устройства ГНСС.

sensorSCIdentifier – идентификатор компонента защиты внешнего устройства ГНСС.

sensorOSIdentifier – идентификатор операционной системы внешнего устройства ГНСС.

2.135. SensorExternalGNSSInstallation

Второе поколение:

Информация, хранящаяся на внешнем устройстве ГНСС, которая относится к установке внешнего датчика ГНСС (приложение 1С, требование 123).

```
SensorExternalGNSSInstallation ::= SEQUENCE {  
    sensorCouplingDateFirst     SensorGNSSCouplingDate,  
    firstVuApprovalNumber       VuApprovalNumber,  
    firstVuSerialNumber         VuSerialNumber,  
    sensorCouplingDateCurrent   SensorGNSSCouplingDate,  
    currentVuApprovalNumber     VuApprovalNumber,  
    currentVUSerialNumber       VuSerialNumber  
}
```

sensorCouplingDateFirst – дата первого соединения внешнего устройства ГНСС с бортовым устройством.

firstVuApprovalNumber – номер официального утверждения первого бортового устройства, соединённого с внешним устройством ГНСС.

firstVuSerialNumber – серийный номер первого бортового устройства, соединённого с внешним устройством ГНСС.

sensorCouplingDateCurrent – дата текущего соединения внешнего устройства ГНСС с бортовым устройством.

currentVuApprovalNumber – номер официального утверждения бортового устройства, в настоящее время соединённого с внешним устройством ГНСС.

currentVUSerialNumber – серийный номер бортового устройства, в настоящее время соединённого с внешним устройством ГНСС.

2.136. SensorExternalGNSSOSIdentifier

Второе поколение:

Идентификатор операционной системы внешнего устройства ГНСС.

```
SensorOSIdentifier ::= IA5String(SIZE(2))
```

Присвоение значения: связано с конкретным производителем.

2.137. SensorExternalGNSSSCIdentifier

Второе поколение:

Данный тип используется, например, для идентификации криптографического модуля внешнего устройства ГНСС.

Идентификатор компонента защиты внешнего устройства ГНСС.

```
SensorExternalGNSSSCIdentifier ::= IA5String(SIZE(8))
```

Присвоение значения: связано с конкретным производителем компонента.

2.138. SensorGNSSCouplingDate

Второе поколение:

Дата соединения внешнего устройства ГНСС с бортовым устройством.

```
SensorGNSSCouplingDate ::= TimeReal
```

Присвоение значения: Не указано.

2.139. SensorGNSSSerialNumber

Второе поколение:

Данный тип используется для хранения серийного номера приёмника ГНСС, и когда он внутри БУ, и когда снаружи него.

Серийный номер приёмника ГНСС.

```
SensorGNSSSerialNumber ::= ExtendedSerialNumber
```

2.140. SensorIdentification

Информация, записанная в датчике движения, которая относится к идентификации датчика движения (приложение 1В, требование 077 и приложение 1С, требование 95).

```
SensorIdentification ::= SEQUENCE {  
    sensorSerialNumber      SensorSerialNumber,  
    sensorApprovalNumber    SensorApprovalNumber,  
    sensorSCIdentifier       SensorSCIdentifier,  
    sensorOSIdentifier       SensorOSIdentifier  
}
```

sensorSerialNumber – расширенный серийный номер датчика движения (включая номер детали и код производителя).

sensorApprovalNumber – номер официального утверждения датчика движения.

sensorSCIdentifier – идентификатор компоненты защиты датчика движения.

sensorOSIdentifier – идентификатор операционной системы датчика движения.

2.141. SensorInstallation

Информация, записанная в датчике движения, которая относится к установке датчика движения (приложение 1В, требование 099 и приложение 1С, требование 122).

```
SensorInstallation ::= SEQUENCE {
    sensorPairingDateFirst      SensorPairingDate,
    firstVuApprovalNumber      VuApprovalNumber,
    firstVuSerialNumber        VuSerialNumber,
    sensorPairingDateCurrent    SensorPairingDate,
    currentVuApprovalNumber     VuApprovalNumber,
    currentVUSerialNumber      VuSerialNumber
}
```

sensorPairingDateFirst – дата первого соединения датчика движения с бортовым устройством.

firstVuApprovalNumber – номер официального утверждения первого бортового устройства, соединённого с датчиком движения.

firstVuSerialNumber – серийный номер первого бортового устройства, соединённого с датчиком движения.

sensorPairingDateCurrent – дата текущего соединения датчика движения с бортовым устройством.

currentVuApprovalNumber – номер официального утверждения бортового устройства, в настоящее время соединённого с датчиком движения.

currentVUSerialNumber – серийный номер бортового устройства, в настоящее время соединённого с датчиком движения.

2.142. SensorInstallationSecData

Информация, записанная на карточке мастерской, которая имеет отношение к данным о защите, необходимым для подсоединения датчиков движения к бортовым устройствам (приложение 1С, требования 308 и 331).

Первое поколение:

```
SensorInstallationSecData ::= TDesSessionKey
```

Присвоение значения: в соответствии с ISO 16844-3.

Второе поколение:

Как описано в приложении 11, на карточке мастерской хранится до трёх ключей для соединения БУ и датчика движения. У этих ключей разные версии.

```
SensorInstallationSecData ::= SEQUENCE {
    kMWCKey1      KMWCKey,
    kMWCKey2      KMWCKey OPTIONAL,
    kMWCKey3      KMWCKey OPTIONAL
}
```

2.143. SensorOSIdentifier

Идентификатор операционной системы датчика движения.

```
SensorOSIdentifier ::= IA5String(SIZE(2))
```

Присвоение значения: связано с конкретным производителем.

2.144. SensorPaired

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к идентификации датчика движения, соединённого с бортовым устройством (приложение 1В, требование 079).

```
SensorPaired ::= SEQUENCE {  
    sensorSerialNumber      SensorSerialNumber,  
    sensorApprovalNumber    SensorApprovalNumber,  
    sensorPairingDateFirst  SensorPairingDate  
}
```

sensorSerialNumber – серийный номер датчика движения, в настоящее время соединённого с бортовым устройством.

sensorApprovalNumber – номер официального утверждения датчика движения, в настоящее время соединённого с бортовым устройством.

sensorPairingDateFirst – дата первого соединения с бортовым устройством датчика движения, в настоящее время соединённого с бортовым устройством.

2.145. SensorPairedRecord

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к идентификации датчика движения, соединённого с бортовым устройством (приложение 1С, требование 97).

```
SensorPairedRecord ::= SEQUENCE {  
    sensorSerialNumber      SensorSerialNumber,  
    sensorApprovalNumber    SensorApprovalNumber,  
    sensorPairingDate       SensorPairingDate  
}
```

sensorSerialNumber – серийный номер датчика движения, соединённого с бортовым устройством.

sensorApprovalNumber – номер официального утверждения датчика движения.

sensorPairingDate – дата соединения датчика движения с бортовым устройством.

2.146. SensorPairingDate

Дата соединения датчика движения с бортовым устройством.

```
SensorPairingDate ::= TimeReal
```

Присвоение значения: Не указано.

2.147. SensorSCIdentifier

Идентификатор компонента защиты датчика движения.

```
SensorSCIdentifier ::= IA5String(SIZE(8))
```

Присвоение значения: связано с конкретным производителем компонента.

2.148. SensorSerialNumber

Серийный номер датчика движения.

```
SensorSerialNumber ::= ExtendedSerialNumber
```

2.149. Signature

Цифровая подпись.

Первое поколение:

```
Signature ::= OCTET STRING (SIZE(128))
```

Присвоение значения: в соответствии с приложением 11 об общих механизмах защиты.

Второе поколение:

```
Signature ::= OCTET STRING (SIZE(64..132))
```

Присвоение значения: в соответствии с приложением 11 об общих механизмах защиты.

2.150. SignatureRecordArray

Второе поколение:

Набор подписей и метаданные, используемые в протоколе загрузки.

```
SignatureRecordArray ::= SEQUENCE {  
    recordType           RecordType,  
    recordSize           INTEGER(1..65535),  
    noOfRecords          INTEGER(0..65535),  
    records              SET SIZE(noOfRecords) OF Signature  
}
```

recordType указывает на тип записи (Signature). **Присвоение значения:** см. RecordType

recordSize – размер Signature в байтах.

noOfRecords – число записей в массиве. Значение устанавливается на 1, так как подписи могут быть разной длины.

records – набор подписей.

2.151. SimilarEventsNumber

Число похожих событий в течение одного конкретного дня (приложение 1B, требование 094 и приложение 1C, требование 117).

```
SimilarEventsNumber ::= INTEGER(0..255)
```

Присвоение значения: 0 не используется, 1 означает, что в данный день имело место и было зарегистрировано только одно событие этого типа, 2 означает, что в этот день имели место 2 события (из которых было сохранено только одно), ...255 означает, что в данный день произошло 255 или более событий этого типа.

2.152. SpecificConditionRecord

Информация, записанная на карточке водителя или мастерской или в бортовом устройстве, которая имеет отношение к особому условию (приложение 1C, требования 130, 276, 301, 328 и 355).

```
SpecificConditionRecord ::= SEQUENCE {  
    entryTime           TimeReal,  
    specificConditionType SpecificConditionType  
}
```

entryTime – дата и время ввода.

specificConditionType – код, позволяющий идентифицировать особое условие.

2.153. SpecificConditions

Информация, записанная на карточке водителя или мастерской или в бортовом устройстве, которая имеет отношение к особому условию (приложение 1С, требования 131, 277, 302, 329 и 356).

Второе поколение:

```
SpecificConditions := SEQUENCE {  
    conditionPointerNewestRecord    INTEGER(0..NoOfSpecificConditionRecords-  
                                     1),  
    specificConditionRecords        SET SIZE(NoOfSpecificConditionRecords)  
                                     OF SpecificConditionRecord  
}
```

conditionPointerNewestRecord – индекс последней обновлённой записи об особых условиях.

Присвоение значения: Число, соответствующее числовому показателю записи данных об особых условиях, которое начинается с 0 в случае первой регистрации регистрации записей, касающихся особых условий, в структуре.

specificConditionRecords – совокупность записей, содержащих информацию об зафиксированных особых условиях.

2.154. SpecificConditionType

Код, позволяющий идентифицировать особые условия (приложение 1В, требования 050b, 105a, 212a и 230a и приложение 1С, требование 62).

```
SpecificConditionType ::= INTEGER(0..255)
```

Первое поколение:

Присвоение значения:

'00'Н	RFU
'01'Н	Вне области применения – начало
'02'Н	Вне области применения – конец
'03'Н	Паром/ж/д переезд
'04'Н .. 'FF'Н	RFU

Второе поколение:

Присвоение значения:

'00'Н	RFU
'01'Н	Вне области применения – начало
'02'Н	Вне области применения – конец
'03'Н	Паром/ж/д переезд – начало
'04'Н	Паром/ж/д переезд – конец
'05'Н .. 'FF'Н	RFU

2.155. Speed

Скорость транспортного средства (км/ч).

```
Speed ::= INTEGER(0..255)
```

Присвоение значения: км в час в рабочем диапазоне от 0 до 220 км/ч.

2.156. SpeedAuthorised

Максимальная разрешённая скорость транспортного средства (определение hh).

```
SpeedAuthorised ::= Speed
```

2.157. SpeedAverage

Средняя скорость за предварительно определённый промежуток времени (км/ч).

```
SpeedAverage ::= Speed
```

2.158. SpeedMax

Максимальная скорость за предварительно определённый промежуток времени.

```
SpeedMax ::= Speed
```

2.159. TachographPayload

Второе поколение:

Определение данного типа данных см. в приложении 14.

2.160. TachographPayloadEncrypted

Второе поколение:

Зашифрованные данные тахографа DER-TLV, т.е. данные, передаваемые в зашифрованном виде сообщением RTM. По вопросу шифрования см. приложение 11, часть Б, глава 13.

```
TachographPayloadEncrypted ::= SEQUENCE {
    tag                OCTET STRING (SIZE (1)),
    length             OCTET STRING (SIZE (1..2)),
    paddingContentIndicatorByte OCTET STRING (SIZE (1)),
    encryptedData      OCTET STRING (SIZE (16..192))
}
```

tag – часть кодирования DER-TLV, которая устанавливается как '87' (см. приложение 11, часть Б, глава 13).

length – часть кодирования DER-TLV, кодирующая длину следующих позиций paddingContentIndicatorByte и encryptedData.

paddingContentIndicatorByte устанавливается на '00'.

encryptedData – это зашифрованная позиция tachographPayload, как указано в приложении 11, часть Б, глава 13. Октетная длина данного элемента данных всегда кратна 16.

2.161. TDesSessionKey

Первое поколение:

Тройной ключ сеанса в системе DES.

```
TDesSessionKey ::= SEQUENCE {
    tDesKeyA          OCTET STRING (SIZE (8)),
    tDesKeyB          OCTET STRING (SIZE (8))
}
```

Присвоение значения: дополнительно не указывается.

2.162. TimeReal

Код совмещённого поля данных даты и времени, в котором дата и время выражаются в секундах, начиная с 00 ч. 00 м. 00 с. 1 января 1970 г. (GMT).

```
TimeReal{INTEGER:TimeRealRange} ::= INTEGER(0..TimeRealRange)
```

Присвоение значения – выровненный октет: Число секунд с полуночи 1 января 1970 г. (GMT).

Максимально возможное отображение даты/времени – 2106 г.

2.163. TyreSize

Обозначение размера шин.

TyreSize ::= IA5String(SIZE(15))

Присвоение значения: в соответствии с Директивой 92/23 (ЕЭС), 31/03/1992, ОЖ L 129, стр. 95.

2.164. VehicleIdentificationNumber

Идентификационный номер транспортного средства (VIN), указывающий на транспортное средство в целом; обычно это серийный номер шасси или номер рамы.

VehicleIdentificationNumber ::= IA5String(SIZE(17))

Присвоение значения: Как описано в ISO 3779.

2.165. VehicleIdentificationNumberRecordArray

Второе поколение:

Идентификационный номер транспортного средства и метаданные, используемые в протоколе загрузки.

```
VehicleIdentificationNumberRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF
                       VehicleIdentificationNumber
}
```

recordType указывает на тип записи (VehicleIdentificationNumber). **Присвоение значения:** см. RecordType

recordSize – размер VehicleIdentificationNumber в байтах.

noOfRecords – число записей в массиве.

records – набор идентификационных номеров транспортных средств.

2.166. VehicleRegistrationIdentification

Идентификационные данные транспортного средства, уникальные для Европы (VRN и государство-член).

```
VehicleRegistrationIdentification ::= SEQUENCE {
    vehicleRegistrationNation    NationNumeric,
    vehicleRegistrationNumber    VehicleRegistrationNumber
}
```

vehicleRegistrationNation – страна, в которой зарегистрировано транспортное средство.

vehicleRegistrationNumber – регистрационный номер транспортного средства (VRN).

2.167. VehicleRegistrationNumber

Регистрационный номер транспортного средства (VRN). Номер регистрации присваивается компетентным органом, регистрирующим транспортное средство.

```
VehicleRegistrationNumber ::= SEQUENCE {
    codePage          ЦЕЛОЕ ЧИСЛО (0..255),
    vehicleRegNumber  OCTET STRING (SIZE(13))
}
```

codePage определяет набор символов, указанный в главе 4,

vehicleRegNumber – VRN, закодированный с помощью указанного набора символов.

Присвоение значения: связано с конкретной страной.

2.168. VehicleRegistrationNumberRecordArray

Второе поколение:

Регистрационный номер транспортного средства и метаданные, используемые в протоколе загрузки.

```
VehicleRegistrationNumberRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF
                        VehicleRegistrationNumber
}
```

recordType указывает на тип записи (VehicleRegistrationNumber). **Присвоение значения:** см. RecordType

recordSize – размер VehicleRegistrationNumber в байтах.

noOfRecords – число записей в массиве.

records – набор регистрационных номеров транспортных средств.

2.169. VuAbility

Второе поколение:

Информация, хранящаяся в БУ о способности или неспособности БУ использовать карточки тахографов первого поколения (приложение 1С, требование 121).

```
VuAbility ::= OCTET STRING (SIZE(1))
```

Присвоение значения – выровненный октет: ‘xxxxxxa’В (8 бит)

Относительно возможности поддерживать карточки первого поколения:

‘a’В	Способность поддерживать карточки тахографов первого поколения:
‘0’ В	Первое поколение поддерживается,
‘1’ В	Первое поколение не поддерживается,

‘xxxxxx’В RFU

2.170. VuActivityDailyData

Первое поколение:

Информация, записанная в БУ, которая имеет отношение к изменению вида деятельности и/или изменению статуса управления и/или изменению состояния карточки за данный календарный день (приложение 1В, требование 084 и приложение 1С, требования 105, 106 и 107) и к состоянию считывающих устройств на 00:00 часов в указанный день.

```
VuActivityDailyData ::= SEQUENCE {
    noOfActivityChanges    INTEGER SIZE(0..1440),
    activityChangeInfos    SET SIZE(noOfActivityChanges) OF
                        ActivityChangeInfo
}
```

noOfActivityChanges – число слов позиции ActivityChangeInfo в совокупности данных activityChangeInfos.

activityChangeInfos – совокупность слов позиции ActivityChangeInfo, записанных в БУ за данный день. Всегда включает в себя два слова ActivityChangeInfo, указывающих на состояние двух считывающих устройств в 00:00 часов в указанный день.

2.171. VuActivityDailyRecordArray

Второе поколение:

Информация, записанная в БУ, которая имеет отношение к изменению вида деятельности и/или изменению статуса управления и/или изменению состояния карточки за данный календарный день (приложение 1С, требования 105, 106 и 107) и к состоянию считывающих устройств на 00:00 часов в указанный день.

```

VuActivityDailyRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize         INTEGER(1..65535),
    noOfRecords        INTEGER(0..65535),
    records            SET SIZE(noOfRecords) OF
                    ActivityChangeInfo
}

```

recordType указывает на тип записи (ActivityChangeInfo). **Присвоение значения:** см. RecordType

recordSize – размер ActivityChangeInfo в байтах.

noOfRecords – число записей в массиве.

activityChangeInfos – совокупность слов позиции ActivityChangeInfo, записанных в БУ за данный день. Всегда включает в себя два слова ActivityChangeInfo, указывающих на состояние двух считывающих устройств в 00:00 часов в указанный день.

2.172. VuApprovalNumber

Номер официального утверждения типа бортового устройства.

Первое поколение:

```
VuApprovalNumber ::= IA5String(SIZE(8))
```

Присвоение значения: Не указано.

Второе поколение:

```
VuApprovalNumber ::= IA5String(SIZE(16))
```

Присвоение значения:

Номер официального утверждения указывается в соответствии с публикацией на соответствующем веб-сайте Европейской комиссии, например, с дефисами, если они есть. Номер официального утверждения смещён влево.

2.173. VuCalibrationData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к калибровке записывающего оборудования (приложение 1B, требование 098).

```

VuCalibrationData ::= SEQUENCE {
    noOfVuCalibrationRecords    INTEGER(0..255),
    vuCalibrationRecords        SET SIZE(noOfVuCalibrationRecords) OF
                                VuCalibrationRecord
}

```

noOfVuCalibrationRecords – число записей, содержащихся в совокупности vuCalibrationRecords.

vuCalibrationRecords – массив записей калибровки.

2.174. VuCalibrationRecord

Информация, записанная в бортовом устройстве, которая относится калибровка записывающего оборудования (приложение 1B, требование 098 и приложение 1C, требования 119 и 120).

Первое поколение:

```
VuCalibrationRecord ::= SEQUENCE {
    calibrationPurpose           CalibrationPurpose,
    workshopName                 Name,
    workshopAddress              Address,
    workshopCardNumber           FullCardNumber,
    workshopCardExpiryDate       TimeReal,
    vehicleIdentificationNumber  VehicleIdentificationNumber,
    vehicleRegistrationIdentification VehicleRegistrationIdentification,
    wVehicleCharacteristicConstant W-VehicleCharacteristicConstant,
    kConstantOfRecordingEquipment K-ConstantOfRecordingEquipment,
    lTyreCircumference           L-TyreCircumference,
    tyreSize                     TyreSize,
    authorisedSpeed               SpeedAuthorised,
    oldOdometerValue             OdometerShort,
    newOdometerValue             OdometerShort,
    oldTimeValue                 TimeReal,
    newTimeValue                 TimeReal,
    nextCalibrationDate          TimeReal
}
```

calibrationPurpose – цель калибровки.

workshopName, workshopAddress – название и адрес мастерской.

workshopCardNumber – идентификатор карточки мастерской, используемой во время калибровки.

workshopCardExpiryDate – дата истечения срока действия карточки.

vehicleIdentificationNumber – идентификационный номер транспортного средства (VIN).

vehicleRegistrationIdentification содержит VRN и государство-член регистрации.

wVehicleCharacteristicConstant – характеристический коэффициент транспортного средства.

kConstantOfRecordingEquipment – постоянная величина записывающего оборудования.

lTyreCircumference – фактическая окружность шин колёс.

tyreSize – обозначение габаритов шин транспортного средства.

authorisedSpeed – допустимая скорость транспортного средства.

oldOdometerValue, newOdometerValue – старые и новые показания одометра.

oldTimeValue, newTimeValue – старые и новые значения даты и времени.

nextCalibrationDate – дата следующей калибровки типа, указанного в CalibrationPurpose, которую должен провести компетентный инспекционный орган.

Второе поколение:


```

VuCalibrationRecord ::= SEQUENCE {
    calibrationPurpose           CalibrationPurpose,
    workshopName                 Name,
    workshopAddress              Address,
    workshopCardNumber           FullCardNumber,
    workshopCardExpiryDate      TimeReal,
    vehicleIdentificationNumber  VehicleIdentificationNumber,
    vehicleRegistrationIdentification VehicleRegistrationIdentification,
    wVehicleCharacteristicConstant W-VehicleCharacteristicConstant,
    kConstantOfRecordingEquipment K-ConstantOfRecordingEquipment,
    lTyreCircumference           L-TyreCircumference,
    tyreSize                     TyreSize,
    authorisedSpeed               SpeedAuthorised,
    oldOdometerValue             OdometerShort,
    newOdometerValue             OdometerShort,
    oldTimeValue                 TimeReal,
    newTimeValue                 TimeReal,
    nextCalibrationDate          TimeReal,
    sealDataVu                   SealDataVu
}

```

В дополнение к первому поколению, используется следующий элемент данных:

sealDataVu предоставляет информацию о пломбах, наложенных на различные компоненты транспортного средства.

2.175. VuCalibrationRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к калибровке записывающего оборудования (приложение 1С, требования 119 и 120).

```

VuCalibrationRecordArray ::= SEQUENCE {
    recordType           RecordType,
    recordSize           INTEGER(1..65535),
    noOfRecords          INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF
                        VuCalibrationRecord
}

```

recordType указывает на тип записи (VuCalibrationRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuCalibrationRecord в байтах.

noOfRecords – число записей в массиве.

records – массив записей калибровки.

2.176. VuCardIWData

Первое поколение:

Информация, записанная в бортовом устройстве, которая имеет отношение к циклам ввода карточек водителя или карточек мастерской в бортовое устройство и их извлечения (приложение 1В, требование 081 и приложение 1С, требование 103).

```

VuCardIWData ::= SEQUENCE {
    noOfIWRecords          INTEGER(0..216-1),
    vuCardIWRecords        SET SIZE(noOfIWRecords) OF
                        VuCardIWRecord
}

```

noOfIWRecords – число записей в совокупности vuCardIWRecords.

vuCardIWRecords – совокупность записей, относящихся к циклам ввода и извлечения карточек.

2.177. VuCardIWRecord

Информация, записанная в бортовом устройстве, которая имеет отношение к циклу ввода карточки водителя или карточки мастерской в бортовое устройство и их извлечения (приложение 1B, требование 081 и приложение 1C, требование 102).

Первое поколение:

```
VuCardIWRecord ::= SEQUENCE {
    cardHolderName           HolderName,
    fullCardNumber           FullCardNumber,
    cardExpiryDate           TimeReal,
    cardInsertionTime        TimeReal,
    vehicleOdometerValueAtInsertion OdometerShort,
    cardSlotNumber           CardSlotNumber,
    cardWithdrawalTime       TimeReal,
    vehicleOdometerValueAtWithdrawal OdometerShort,
    previousVehicleInfo      PreviousVehicleInfo,
    manualInputFlag          ManualInputFlag
}
```

cardHolderName – фамилия и имя (имена) владельца карточки водителя или мастерской, записанные на карточке.

fullCardNumber – тип карточки, выдавшее её государство-член и номер карточки, записанные на карточке.

cardExpiryDate – дата истечения срока действия карточки, записанная на карточке.

cardInsertionTime – дата и время ввода.

vehicleOdometerValueAtInsertion – показание одометра транспортного средства в момент ввода карточки.

cardSlotNumber – считывающее устройство, в которое вставляется карточка.

cardWithdrawalTime – дата и время извлечения.

vehicleOdometerValueAtWithdrawal – показание одометра транспортного средства в момент извлечения карточки.

previousVehicleInfo – информация о предыдущем транспортном средстве, которым пользовался водитель, записанная на карточке.

manualInputFlag – метка, позволяющая определить, ввёл ли владелец карточки в момент её ввода данные о деятельности водителя вручную.

Второе поколение:

```
VuCardIWRecord ::= SEQUENCE {
    cardHolderName           HolderName,
    fullCardNumberAndGeneration FullCardNumberAndGeneration,
    cardExpiryDate           TimeReal,
    cardInsertionTime        TimeReal,
    vehicleOdometerValueAtInsertion OdometerShort,
    cardSlotNumber           CardSlotNumber,
    cardWithdrawalTime       TimeReal,
    vehicleOdometerValueAtWithdrawal OdometerShort,
    previousVehicleInfo      PreviousVehicleInfo,
    manualInputFlag          ManualInputFlag
}
```

Вместо **fullCardNumber** в структуре данных второго поколения используется следующий элемент данных.

fullCardNumberAndGeneration – тип карточки, выдавшее её государство-член и номер и поколение карточки, записанные на карточке.

2.178. VuCardIWRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая имеет отношение к циклам ввода карточек водителя или карточек мастерской в бортовое устройство и их извлечения (приложение 1С, требование 103).

```
VuCardIWRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF VuCardIWRecord
}
```

recordType указывает на тип записи (VuCardIWRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuCardIWRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей, относящихся к циклам ввода и извлечения карточек.

2.179. VuCardRecord

Второе поколение:

Информация, записанная в бортовом устройстве, об используемой карточке тахографа (приложение 1С, требование 132).

```
VuCardRecord ::= SEQUENCE {
    cardExtendedSerialNumber ExtendedSerialNumber,
    cardPersonaliserID       OCTET STRING(SIZE(1)),
    typeofTachographCardID   EquipmentType,
    cardStructureVersion      CardStructureVersion,
    cardNumber                CardNumber
}
```

cardExtendedSerialNumber считывается из файла EF_ICC в MF карточки.

cardPersonaliserID считывается из файла EF_ICC в MF карточки.

typeofTachographCardId считывается из файла EF_Application_Identification в DF_Tachograph_G2

cardStructureVersion считывается из файла EF_Application_Identification в DF_Tachograph_G2.

cardNumber считывается из файла EF_Identification в DF_Tachograph_G2.

2.180. VuCardRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, об карточках тахографа, используемых в данном бортовом устройстве. Данная информация включается в анализ БУ, касающийся проблем с карточками (приложение 1С, требование 132).

```
VuCardRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF VuCardRecord
}
```

recordType указывает на тип записи (VuCardRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuCardRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей, относящихся к карточкам тахографа, используемым в БУ.

2.181. VuCertificate

Сертификат открытого ключа бортового устройства.

```
VuCertificate ::= Certificate
```

2.182. VuCertificateRecordArray

Второе поколение:

Сертификат БУ и метаданные, используемые в протоколе загрузки.

```
VuCertificateRecordArray ::= SEQUENCE {  
    recordType          RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records              SET SIZE(noOfRecords) OF VuCertificate  
}
```

recordType указывает на тип записи (VuCertificate). **Присвоение значения:** см. RecordType

recordSize – размер VuCertificate в байтах.

noOfRecords – число записей в массиве. Значение устанавливается на 1, так как сертификаты могут быть разной длины.

records – набор сертификатов БУ.

2.183. VuCompanyLocksData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к блокировке предприятием (приложение 1В, требование 104).

```
VuCompanyLocksData ::= SEQUENCE {  
    noOfLocks           INTEGER(0..255),  
    vuCompanyLocksRecords SET SIZE(noOfLocks) OF  
                          VuCompanyLocksRecord  
}
```

noOfLocks – число блокировок, перечисленных в файле vuCompanyLocksRecords

vuCompanyLocksRecords – совокупность записей действий блокировки, произведённых предприятием.

2.184. VuCompanyLocksRecord

Информация, записанная в бортовом устройстве, которая относится к одной блокировке предприятием (приложение 1В, требование 104 и приложение 1С, требование 128).

Первое поколение:

```
VuCompanyLocksRecord ::= SEQUENCE {  
    lockInTime          TimeReal,  
    lockOutTime         TimeReal,  
    companyName         Name,  
    companyAddress      Address,  
    companyCardNumber   FullCardNumber  
}
```

lockInTime, lockOutTime – дата и время блокировки и снятия блокировки.

companyName, companyAddress – название и адрес предприятия, которое произвело блокировку.

companyCardNumber – идентификационные данные карточки, используемой для блокировки.

Второе поколение:

```

VuCompanyLocksRecord ::= SEQUENCE {
    lockInTime           TimeReal,
    lockOutTime          TimeReal,
    companyName          Name,
    companyAddress       Address,
    companyCardNumberAndGeneration FullCardNumberAndGeneration
}

```

Вместо `companyCardNumber` в структуре данных второго поколения используется следующий элемент данных.

companyCardNumberAndGeneration идентифицирует карточку, используемую при блокировке, и её поколение.

2.185. VuCompanyLocksRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к блокировке предприятием (приложение 1С, требование 128).

```

VuCompanyLocksRecordArray ::= SEQUENCE {
    recordType           RecordType,
    recordSize           INTEGER(1..65535),
    noOfRecords          INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF
                        VuCompanyLocksRecord
}

```

recordType указывает на тип записи (`VuCompanyLocksRecord`). **Присвоение значения:** см. `RecordType`

recordSize – размер `VuCompanyLocksRecord` в байтах.

noOfRecords – число записей в массиве. Значение 0..255.

records – совокупность записей действий блокировки, произведённых предприятием.

2.186. VuControlActivityData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к контрольным действиям, проведённым с данным БУ (приложение 1В, требование 102).

```

VuControlActivityData ::= SEQUENCE {
    noOfControls         INTEGER(0..20),
    vuControlActivityRecords SET SIZE(noOfControls) OF
                        VuControlActivityRecord
}

```

noOfControls – число контрольных действий, перечисленных в `vuControlActivityRecords`.

vuControlActivityRecords – совокупность записей контрольных действий.

2.187. VuControlActivityRecord

Информация, записанная в бортовом устройстве, которая относится к контрольному действию с данным БУ (приложение 1В, требование 102 и приложение 1С, требование 126).

Первое поколение:

```
VuControlActivityRecord ::= SEQUENCE {
    controlType          ControlType,
    controlTime          TimeReal,
    controlCardNumber    FullCardNumber,
    downloadPeriodBeginTime TimeReal,
    downloadPeriodEndTime TimeReal
}
```

controlType – тип контроля.

controlTime – дата и время ввода.

controlCardNumber – идентификатор карточки контролёра, использованной для контрольных действий.

downloadPeriodBeginTime – время начала периода, за который загружаются данные (в случае загрузки).

downloadPeriodEndTime – время окончания периода, за который загружаются данные (в случае загрузки).

Второе поколение:

```
VuControlActivityRecord ::= SEQUENCE {
    controlType          ControlType,
    controlTime          TimeReal,
    controlCardNumberAndGeneration FullCardNumberAndGeneration,
    downloadPeriodBeginTime TimeReal,
    downloadPeriodEndTime TimeReal
}
```

Вместо controlCardNumber в структуре данных второго поколения используется следующий элемент данных.

controlCardNumberAndGeneration – идентификатор контрольной карточки, использованной для контрольных действий.

2.188. VuControlActivityRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к контрольным действиям, проведённым с данным БУ (приложение 1С, требование 126).

```
VuControlActivityRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF
                        VuControlActivityRecord
}
```

recordType указывает на тип записи (VuControlActivityRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuControlActivityRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей контрольных действий, связанных с БУ.

2.189. VuDataBlockCounter

Счётчик, записанный на карточке, который позволяет определять последовательную нумерацию циклов ввода и извлечения карточки в бортовых устройствах.

```
VuDataBlockCounter ::= BCDString(SIZE(2))
```

Присвоение значения: Порядковый номер с макс. значением 9 999, который снова начинается с 0.

2.190. VuDetailedSpeedBlock

Информация, записанная в бортовом устройстве, которая относится к подробным данным скорости транспортного средства за минуту, в течение которой транспортное средство находится в движении (приложение 1В, требование 093 и приложение 1С, требование 116).

```
VuDetailedSpeedBlock ::= SEQUENCE {
    speedBlockBeginDate TimeReal,
    speedsPerSecond      SEQUENCE SIZE(60) OF Speed
}
```

speedBlockBeginDate – дата и время первого значения скорости в блоке данных.

speedsPerSecond – хронологическая последовательность измеряемых скоростей за каждую секунду в течение минуты, которая начинает отсчитываться с момента времени позиции speedBlockBeginDate (включительно).

2.191. VuDetailedSpeedBlockRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к подробным данным о скорости транспортного средства.

```
VuDetailedSpeedBlockRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords        INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF
                        VuDetailedSpeedBlock
}
```

recordType указывает на тип записи (VuDetailedSpeedBlock). **Присвоение значения:** см. RecordType

recordSize – размер VuDetailedSpeedBlock в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей о подробных блоках данных скорости.

2.192. VuDetailedSpeedData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к подробным данным о скорости транспортного средства.

```
VuDetailedSpeedData ::= SEQUENCE {
    noOfSpeedBlocks    INTEGER(0..216-1),
    vuDetailedSpeedBlocks SET SIZE(noOfSpeedBlocks) OF
                        VuDetailedSpeedBlock
}
```

noOfSpeedBlocks – число блоков данных скорости в совокупности данных vuDetailedSpeedBlocks.

vuDetailedSpeedBlocks – совокупность записей о подробных блоках данных скорости.

2.193. VuDownloadablePeriod

Самая давняя и новейшая даты, на которые в бортовом устройстве хранятся данные, связанные с деятельностью водителя (приложение 1В, требования 081, 084 или 087 и приложение 1С, требования 102, 105, 108).

```
VuDownloadablePeriod ::= SEQUENCE {
    minDownloadableTime      TimeReal
    maxDownloadableTime      TimeReal
}
```

minDownloadableTime – самая давний ввод карточки или изменение вида деятельности, или дата и время прибытия на место, хранящиеся в БУ.

maxDownloadableTime – последнее извлечение карточки или изменение вида деятельности, или дата и время прибытия на место, хранящиеся в БУ.

2.194. VuDownloadablePeriodRecordArray

Второе поколение:

VUDownloadablePeriod и метаданные, используемые в протоколе загрузки.

```
VuDownloadablePeriodRecordArray ::= SEQUENCE {
    recordType                RecordType,
    recordSize                INTEGER(1..65535),
    noOfRecords              INTEGER(0..65535),
    records                   SET SIZE(noOfRecords) OF
                             VuDownloadablePeriod
}
```

recordType указывает на тип записи (VuDownloadablePeriod). **Присвоение значения:** см. RecordType

recordSize – размер VuDownloadablePeriod в байтах.

noOfRecords – число записей в массиве.

records – массив записей VuDownloadablePeriod.

2.195. VuDownloadActivityData

Информация, записанная в бортовом устройстве, которая относится к последней загрузке (приложение 1В, требование 105 и приложение 1С, требование 129).

Первое поколение:

```
VuDownloadActivityData ::= SEQUENCE {
    downloadingTime          TimeReal,
    fullCardNumber           FullCardNumber,
    companyOrWorkshopName    Name
}
```

downloadingTime – дата и время загрузки.

fullCardNumber – идентификатор использованной карточки, разрешающей загрузку.

companyOrWorkshopName – название предприятия или мастерской.

Второе поколение:


```
VuDownloadActivityData ::= SEQUENCE {
    downloadingTime          TimeReal,
    fullCardNumberAndGeneration FullCardNumberAndGeneration,
    companyOrWorkshopName    Name
}
```

Вместо `fullCardNumber` в структуре данных второго поколения используется следующий элемент данных.

fullCardNumberAndGeneration – идентификатор использованной карточки, разрешающей загрузку, и её поколения.

2.196. VuDownloadActivityDataRecordArray

Второе поколение:

Информация, связанная с последней загрузкой данных с БУ (приложение 1С, требование 129).

```
VuDownloadActivityDataRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF
                        VuDownloadActivityData
}
```

recordType указывает на тип записи (`VuDownloadActivityData`). **Присвоение значения:** см. `RecordType`

recordSize – размер `VuDownloadActivityData` в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей действий загрузки данных.

2.197. VuEventData

Первое поколение:

Информация, записанная в бортовом устройстве, связанная с событиями (приложение 1В, требование 094, кроме события превышения скорости).

```
VuEventData ::= SEQUENCE {
    noOfVuEvents          INTEGER(0..255),
    vuEventRecords        SET SIZE(noOfVuEvents) OF VuEventRecord
}
```

noOfVuEvents – число событий, перечисленных в совокупности `vuEventRecords`.

vuEventRecords – массив записей о событиях.

2.198. VuEventRecord

Информация, записанная в бортовом устройстве, связанная с событием (приложение 1В, требование 094 и приложение 1С, требование 117, кроме события превышения скорости).

Первое поколение:

```
VuEventRecord ::= SEQUENCE {
    eventType                EventFaultType,
    eventRecordPurpose       EventFaultRecordPurpose,
    eventBeginTime           TimeReal,
    eventEndTime             TimeReal,
    cardNumberDriverSlotBegin FullCardNumber,
    cardNumberCodriverSlotBegin FullCardNumber,
    cardNumberDriverSlotEnd FullCardNumber,
    cardNumberCodriverSlotEnd FullCardNumber,
    similarEventsNumber      SimilarEventsNumber
}
```

eventType – тип события.

eventRecordPurpose – цель регистрации данного события.

eventBeginTime – дата и время начала события.

eventEndTime – дата и время завершения события.

cardNumberDriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство водителя в начале события.

cardNumberCodriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство второго водителя в начале события.

cardNumberDriverSlotEnd – идентификатор карточки, вставленной в считывающее устройство водителя в конце события.

cardNumberCodriverSlotEnd – идентификатор карточки, вставленной в считывающее устройство второго водителя в конце события.

similarEventsNumber – число аналогичных событий за указанный день.

Эта последовательность может быть использована для всех событий, помимо случаев превышения скорости.

Второе поколение:

```
VuEventRecord ::= SEQUENCE {
    eventType                EventFaultType,
    eventRecordPurpose       EventFaultRecordPurpose,
    eventBeginTime           TimeReal,
    eventEndTime             TimeReal,
    cardNumberAndGenDriverSlotBegin FullCardNumberAndGeneration,
    cardNumberAndGenCodriverSlotBegin FullCardNumberAndGeneration,
    cardNumberAndGenDriverSlotEnd FullCardNumberAndGeneration,
    cardNumberAndGenCodriverSlotEnd FullCardNumberAndGeneration,
    similarEventsNumber      SimilarEventsNumber,
    manufacturerSpecificEventFaultData ManufacturerSpecificEventFaultData
}
```

В дополнение к первому поколению, используются следующие элементы данных:

manufacturerSpecificEventFaultData содержит дополнительную информацию, связанную с конкретным производителем, о событии.

Вместо `cardNumberDriverSlotBegin`, `cardNumberCodriverSlotBegin`, `cardNumberDriverSlotEnd` и `cardNumberCodriverSlotEnd` в структуре данных второго поколения используются следующие элементы данных:

`cardNumberAndGenDriverSlotBegin` – идентификатор карточки, вставленной в считывающее устройство водителя в начале события, и её поколения.

`cardNumberAndGenCodriverSlotBegin` – идентификатор карточки, вставленной в считывающее устройство второго водителя в начале события, и её поколения.

`cardNumberAndGenDriverSlotEnd` – идентификатор карточки, вставленной в считывающее устройство водителя в конце события, и её поколения.

`cardNumberAndGenCodriverSlotEnd` – идентификатор карточки, вставленной в считывающее устройство второго водителя в конце события, и её поколения.

Если событие представляет собой нестыковку во времени, `eventBeginTime` и `eventEndTime` следует толковать следующим образом:

`eventBeginTime` – дата и время в записывающем оборудовании.

`eventEndTime` – дата и время по ГНСС.

2.199. VuEventRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, связанная с событиями (приложение 1С, требование 117, кроме события превышения скорости).

```
VuEventRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF VuEventRecord
}
```

`recordType` указывает на тип записи (`VuEventRecord`). **Присвоение значения:** см. `RecordType`

`recordSize` – размер `VuEventRecord` в байтах.

`noOfRecords` – число записей в массиве.

`records` – массив записей о событиях.

2.200. VuFaultData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к неисправностям (приложение 1В, требование 096).

```
VuFaultData ::= SEQUENCE {
    noOfVuFaults        INTEGER(0..255),
    vuFaultRecords      SET SIZE(noOfVuFaults) OF VuFaultRecord
}
```

`noOfVuFaults` – число неисправностей, перечисленных в совокупности `vuFaultRecords`.

`vuFaultRecords` – массив записей о неисправностях.

2.201. VuFaultRecord

Информация, записанная в бортовом устройстве, которая относится к неисправности (приложение 1В, требование 096 и приложение 1С, требование 118).

Первое поколение:

```
VuFaultRecord ::= SEQUENCE {
    faultType                EventFaultType,
    faultRecordPurpose       EventFaultRecordPurpose,
    faultBeginTime           TimeReal,
    faultEndTime             TimeReal,
    cardNumberDriverSlotBegin FullCardNumber,
    cardNumberCodriverSlotBegin FullCardNumber,
    cardNumberDriverSlotEnd  FullCardNumber,
    cardNumberCodriverSlotEnd FullCardNumber
}
```

faultType – тип неисправности записывающего оборудования.

faultRecordPurpose – цель регистрации данной неисправности.

faultBeginTime – дата и время начала неисправности.

faultEndTime – дата и время окончания неисправности.

cardNumberDriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство водителя в начале неисправности.

cardNumberCodriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство второго водителя в начале неисправности.

cardNumberDriverSlotEnd – идентификатор карточки, вставленной в считывающее устройство водителя в конце неисправности.

cardNumberCodriverSlotEnd – идентификатор карточки, вставленной в считывающее устройство второго водителя в конце неисправности.

Второе поколение:

```
VuFaultRecord ::= SEQUENCE {
    faultType                EventFaultType,
    faultRecordPurpose       EventFaultRecordPurpose,
    faultBeginTime           TimeReal,
    faultEndTime             TimeReal,
    cardNumberAndGenDriverSlotBegin FullCardNumberAndGeneration,
    cardNumberAndGenCodriverSlotBegin FullCardNumberAndGeneration,
    cardNumberAndGenDriverSlotEnd  FullCardNumberAndGeneration,
    cardNumberAndGenCodriverSlotEnd FullCardNumberAndGeneration,
    manufacturerSpecificEventFaultData ManufacturerSpecificEventFaultData
}
```

В дополнение к первому поколению, используется следующий элемент данных:

manufacturerSpecificEventFaultData содержит дополнительную информацию, связанную с конкретным производителем, о неисправности.

Вместо **cardNumberDriverSlotBegin**, **cardNumberCodriverSlotBegin**, **cardNumberDriverSlotEnd** и **cardNumberCodriverSlotEnd** в структуре данных второго поколения используются следующие элементы данных:

cardNumberAndGenDriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство водителя в начале неисправности, и её поколения.

cardNumberAndGenCodriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство второго водителя в начале неисправности, и её поколения.

cardNumberAndGenDriverSlotEnd – идентификатор карточки, вставленной в считывающее устройство водителя в конце неисправности, и её поколения.

cardNumberAndGenCodriverSlotEnd – идентификатор карточки, вставленной в считывающее устройство второго водителя в конце неисправности, и её поколения.

2.202. VuFaultRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к неисправностям (приложение 1С, требование 118).

```
VuFaultRecordArray ::= SEQUENCE {  
    recordType           RecordType,  
    recordSize          INTEGER(1..65535),  
    noOfRecords         INTEGER(0..65535),  
    records             SET SIZE(noOfRecords) OF VuFaultRecord  
}
```

recordType указывает на тип записи (VuFaultRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuFaultRecord в байтах.

noOfRecords – число записей в массиве.

records – массив записей о неисправностях.

2.203. VuGNSSCDRecord

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к местоположению транспортного средства по ГНСС, если непрерывное время управления водителем достигает значения, кратного трём часам (приложение 1С, требования 108, 110).

```
VuGNSSCDRecord ::= SEQUENCE {  
    timeStamp           TimeReal,  
    cardNumberAndGenDriverSlot FullCardNumberAndGeneration,  
    cardNumberAndGenCodriverSlot FullCardNumberAndGeneration,  
    gnssPlaceRecord    GNSSPlaceRecord  
}
```

timeStamp – дата и время, когда непрерывное время управления владельцем карточки достигает значения, кратного трём часам.

cardNumberAndGenDriverSlot – идентификатор карточки, вставленной в считывающее устройство водителя, и её поколения.

cardNumberAndGenCodriverSlot – идентификатор карточки, вставленной в считывающее устройство второго водителя, и её поколения.

gnssPlaceRecord содержит информацию, связанную с местоположением транспортного средства.

2.204. VuGNSSCDRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к местоположению транспортного средства по ГНСС, если непрерывное время управления водителем достигает значения, кратного трём часам (приложение 1С, требования 108 и 110).

```

VuGNSSCDRecordArray ::= SEQUENCE {
    recordType           RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF VuGNSSCDRecord
}

```

recordType указывает на тип записи (VuGNSSCDRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuGNSSCDRecord в байтах.

noOfRecords – число записей в массиве.

records – массив записей о непрерывном управлении по ГНСС.

2.205. VuIdentification

Информация, записанная в бортовом устройстве, которая относится к идентификации бортового устройства (приложение 1B, требование 075 и приложение 1C, требования 93 и 121).

Первое поколение:

```

VuIdentification ::= SEQUENCE {
    vuManufacturerName    VuManufacturerName,
    vuManufacturerAddress VuManufacturerAddress,
    vuPartNumber          VuPartNumber,
    vuSerialNumber        VuSerialNumber,
    vuSoftwareIdentification VuSoftwareIdentification,
    vuManufacturingDate   VuManufacturingDate,
    vuApprovalNumber      VuApprovalNumber
}

```

vuManufacturerName – название производителя бортового устройства.

vuManufacturerAddress – адрес производителя бортового устройства.

vuPartNumber – номер детали бортового устройства.

vuSerialNumber – серийный номер бортового устройства.

vuSoftwareIdentification позволяет идентифицировать программное обеспечение, установленное в бортовом устройстве.

vuManufacturingDate – дата изготовления бортового устройства.

vuApprovalNumber – номер официального утверждения типа бортового устройства.

Второе поколение:

```

VuIdentification ::= SEQUENCE {
    vuManufacturerName    VuManufacturerName,
    vuManufacturerAddress VuManufacturerAddress,
    vuPartNumber          VuPartNumber,
    vuSerialNumber        VuSerialNumber,
    vuSoftwareIdentification VuSoftwareIdentification,
    vuManufacturingDate   VuManufacturingDate,
    vuApprovalNumber      VuApprovalNumber,
    vuGeneration          Generation,
    vuAbility              VuAbility
}

```

В дополнение к первому поколению, используются следующие элементы данных:

vuGeneration идентифицирует поколение бортового устройства.

vuAbility предоставляет информацию о том, поддерживает ли БУ карточки тахографов первого поколения или нет.

2.206. VuIdentificationRecordArray

Второе поколение:

VuIdentification и метаданные, используемые в протоколе загрузки.

```
VuIdentificationRecordArray ::= SEQUENCE {
    recordType                RecordType,
    recordSize                INTEGER(1..65535),
    noOfRecords              INTEGER(0..65535),
    records                   SET SIZE(noOfRecords) OF
                             VuIdentification
}
```

recordType указывает на тип записи (VuIdentification). **Присвоение значения:** см. RecordType

recordSize – размер VuIdentification в байтах.

noOfRecords – число записей в массиве.

records – массив записей VuIdentification.

2.207. VuITSConsentRecord

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к согласию водителя на пользование интеллектуальными транспортными системами.

```
VuITSConsentRecord ::= SEQUENCE {
    cardNumberAndGen         FullCardNumberAndGeneration,
    consent                  BOOLEAN
}
```

cardNumberAndGen позволяет идентифицировать карточку и её поколение. Это должна быть карточка водителя или карточка мастерской.

consent – это метка, указывающая на то, дал ли водитель своё согласие на пользование интеллектуальными транспортными системами на своём транспортном средстве/бортовом устройстве.

Присвоение значения:

TRUE означает согласие водителя на пользование интеллектуальными транспортными системами

FALSE означает несогласие водителя на пользование интеллектуальными транспортными системами

2.208. VuITSConsentRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к согласию на пользование интеллектуальными транспортными системами (приложение 1С, требование 200).

```

VuITSConsentRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF
                        VuITSConsentRecord
}

```

recordType указывает на тип записи (VuITSConsentRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuITSConsentRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей о согласии на ИТС.

2.209. VuManufacturerAddress

Адрес производителя бортового устройства.

```
VuManufacturerAddress ::= Address
```

Присвоение значения: Не указано.

2.210. VuManufacturerName

Название производителя бортового устройства.

```
VuManufacturerName ::= Name
```

Присвоение значения: Не указано.

2.211. VuManufacturingDate

Дата изготовления бортового устройства.

```
VuManufacturingDate ::= TimeReal
```

Присвоение значения: Не указано.

2.212. VuOverSpeedingControlData

Информация, записанная в бортовом устройстве, которая относится к событиям превышения скорости с момента последнего контроля (приложение 1В, требование 095 и приложение 1С, требование 117).

```

VuOverSpeedingControlData ::= SEQUENCE {
    lastOverspeedControlTime    TimeReal,
    firstOverspeedSince         TimeReal,
    numberOfOverspeedSince      OverspeedNumber
}

```

lastOverspeedControlTime – дата и время последнего контроля за превышением скорости.

firstOverspeedSince – дата и время первого превышения скорости после данного контроля за превышением скорости.

numberOfOverspeedSince – число случаев превышения скорости после последнего контроля за превышением скорости.

2.213. VuOverSpeedingControlDataRecordArray

Второе поколение:

VuOverSpeedingControlData и метаданные, используемые в протоколе загрузки.


```
VuOverSpeedingControlDataRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF
                        VuOverSpeedingControlData
}

```

recordType указывает на тип записи (VuOverSpeedingControlData). **Присвоение значения:** см. RecordType

recordSize – размер VuOverSpeedingControlData в байтах.

noOfRecords – число записей в массиве.

records – массив записей данных контроля за превышением скорости.

2.214. VuOverSpeedingEventData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к событиям превышения скорости (приложение 1B, требование 094).

```
VuOverSpeedingEventData ::= SEQUENCE {
    noOfVuOverSpeedingEvents    INTEGER(0..255),
    vuOverSpeedingEventRecords SET SIZE(noOfVuOverSpeedingEvents) OF
                                VuOverSpeedingEventRecord
}

```

noOfVuOverSpeedingEvents – число событий, перечисленных в совокупности vuOverSpeedingEventRecords.

vuOverSpeedingEventRecords – совокупность записей о событиях превышения скорости.

2.215. VuOverSpeedingEventRecord

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к событиям превышения скорости (приложение 1B, требование 094 и приложение 1C, требование 117).

```
VuOverSpeedingEventRecord ::= SEQUENCE {
    eventType          EventFaultType,
    eventRecordPurpose EventFaultRecordPurpose,
    eventBeginTime     TimeReal,
    eventEndTime       TimeReal,
    maxSpeedValue      SpeedMax,
    averageSpeedValue  SpeedAverage,
    cardNumberDriverSlotBegin FullCardNumber,
    similarEventsNumber SimilarEventsNumber
}

```

eventType – тип события.

eventRecordPurpose – цель регистрации данного события.

eventBeginTime – дата и время начала события.

eventEndTime – дата и время завершения события.

maxSpeedValue – максимальная скорость, измеренная во время события.

averageSpeedValue – средняя арифметическая скорость, измеренная во время события.

cardNumberDriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство водителя в начале события.

similarEventsNumber – число аналогичных событий за указанный день.

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к событиям превышения скорости (приложение 1В, требование 094 и приложение 1С, требование 117).

```
VuOverSpeedingEventRecord ::= SEQUENCE {
    eventType                EventFaultType,
    eventRecordPurpose       EventFaultRecordPurpose,
    eventBeginTime           TimeReal,
    eventEndTime             TimeReal,
    maxSpeedValue            SpeedMax,
    averageSpeedValue        SpeedAverage,
    cardNumberAndGenDriverSlotBegin FullCardNumberAndGeneration,
    similarEventsNumber      SimilarEventsNumber
}
```

Вместо `cardNumberDriverSlotBegin` в структуре данных второго поколения используется следующий элемент данных:

cardNumberAndGenDriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство водителя в начале события, и её поколения.

2.216. VuOverSpeedingEventRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к событиям превышения скорости (приложение 1С, требование 117).

```
VuOverSpeedingEventRecordArray ::= SEQUENCE {
    recordType                RecordType,
    recordSize                INTEGER(1..65535),
    noOfRecords               INTEGER(0..65535),
    records                   SET SIZE(noOfRecords) OF
                             VuOverSpeedingEventRecord
}
```

recordType указывает на тип записи (`VuOverSpeedingEventRecord`). **Присвоение значения:** см. `RecordType`

recordSize – размер `VuOverSpeedingEventRecord` в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей о событиях превышения скорости.

2.217. VuPartNumber

Номер детали бортового устройства.

```
VuPartNumber ::= IA5String(SIZE(16))
```

Присвоение значения: Относится конкретно к производителю БУ.

2.218. VuPlaceDailyWorkPeriodData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к местам, где водители начинают или заканчивают дневной период работы (приложение 1В, требование 087 и приложение 1С, требования 108 и 110).

```
VuPlaceDailyWorkPeriodData ::= SEQUENCE {
    noOfPlaceRecords          INTEGER(0..255),
    vuPlaceDailyWorkPeriodRecords SET SIZE(noOfPlaceRecords) OF
                                VuPlaceDailyWorkPeriodRecord
}
```

noOfPlaceRecords – число событий, перечисленных в совокупности vuPlaceDailyWorkPeriodRecords.

vuPlaceDailyWorkPeriodRecords – совокупность записей о местах.

2.219. VuPlaceDailyWorkPeriodRecord

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к месту, где водитель начинает или заканчивает дневной период работы (приложение 1В, требование 087 и приложение 1С, требования 108 и 110).

```
VuPlaceDailyWorkPeriodRecord ::= SEQUENCE {
    fullCardNumber            FullCardNumber,
    placeRecord                PlaceRecord
}
```

fullCardNumber – тип карточки водителя, государство-член, выдавшее карточку, и номер карточки.

placeRecord содержит информацию, связанную с местом ввода.

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к месту, где водитель начинает или заканчивает дневной период работы (приложение 1В, требование 087 и приложение 1С, требования 108 и 110).

```
VuPlaceDailyWorkPeriodRecord ::= SEQUENCE {
    fullCardNumberAndGeneration FullCardNumberAndGeneration,
    placeRecord                PlaceRecord
}
```

Вместо fullCardNumber в структуре данных второго поколения используется следующий элемент данных:

fullCardNumberAndGeneration – тип карточки, выдавшее её государство-член и номер и поколение карточки, записанные на карточке.

2.220. VuPlaceDailyWorkPeriodRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к местам, где водители начинают или заканчивают дневной период работы (приложение 1С, требования 108 и 110).

```

VuPlaceDailyWorkPeriodRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF
                        VuPlaceDailyWorkPeriodRecord
}

```

recordType указывает на тип записи (VuPlaceDailyWorkPeriodRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuPlaceDailyWorkPeriodRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей о местах.

2.221. VuPrivateKey

Первое поколение:

Закрытый ключ бортового устройства.

```
VuPrivateKey ::= RSAKeyPrivateExponent
```

2.222. VuPublicKey

Первое поколение:

Открытый ключ бортового устройства.

```
VuPublicKey ::= PublicKey
```

2.223. VuSerialNumber

Серийный номер бортового устройства (приложение 1B, требование 075 и приложение 1C, требование 93).

```
VuSerialNumber ::= ExtendedSerialNumber
```

2.224. VuSoftInstallationDate

Дата установки программного обеспечения бортового устройства.

```
VuSoftInstallationDate ::= TimeReal
```

Присвоение значения: Не указано.

2.225. VuSoftwareIdentification

Информация, записанная в бортовом устройстве, которая относится к установленному программному обеспечению.

```

VuSoftwareIdentification ::= SEQUENCE {
    vuSoftwareVersion      VuSoftwareVersion,
    vuSoftInstallationDate VuSoftInstallationDate
}

```

vuSoftwareVersion – номер версии программного обеспечения бортового устройства.

vuSoftInstallationDate – дата установки версии программного обеспечения.

2.226. VuSoftwareVersion

Номер версии программного обеспечения бортового устройства.

```
VuSoftwareVersion ::= IA5String(SIZE(4))
```

Присвоение значения: Не указано.

2.227. VuSpecificConditionData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к особым условиям.

```
VuSpecificConditionData ::= SEQUENCE {
    noOfSpecificConditionRecords    INTEGER(0..216-1)
    specificConditionRecords        SET SIZE (noOfSpecificConditionRecords)
                                   OF SpecificConditionRecord
}
```

noOfSpecificConditionRecords – число записей, перечисленных в совокупности specificConditionRecords.

specificConditionRecords – совокупность записей об особых условиях.

2.228. VuSpecificConditionRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к особым условиям (приложение 1С, требование 130).

```
VuSpecificConditionRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records             SET SIZE(noOfRecords) OF
                       SpecificConditionRecord
}
```

recordType указывает на тип записи (SpecificConditionRecord). **Присвоение значения:** см. RecordType

recordSize – размер SpecificConditionRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей об особых условиях.

2.229. VuTimeAdjustmentData

Первое поколение:

Информация, записанная в бортовом устройстве, которая относится к корректировке времени, не связанной с регулярной калибровкой (приложение 1В, требование 101).

```
VuTimeAdjustmentData ::= SEQUENCE {
    noOfVuTimeAdjRecords    INTEGER(0..6),
    vuTimeAdjustmentRecords SET SIZE (noOfVuTimeAdjRecords) OF
                             VuTimeAdjustmentRecord
}
```

noOfVuTimeAdjRecords – число записей в vuTimeAdjustmentRecords.

vuTimeAdjustmentRecords – совокупность записей о корректировках времени.

2.230. VuTimeAdjustmentGNSSRecord

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к корректировке времени на основании данных времени по ГНСС (приложение 1С, требования 124 и 125).

```
VuTimeAdjustmentGNSSRecord ::= SEQUENCE {
    oldTimeValue          TimeReal,
    newTimeValue          TimeReal
}
```

oldTimeValue, newTimeValue – старые и новые значения даты и времени.

2.231. VuTimeAdjustmentGNSSRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к корректировке времени на основании данных времени по ГНСС (приложение 1С, требования 124 и 125).

```
VuTimeAdjustmentGNSSRecordArray ::= SEQUENCE {
    recordType            RecordType,
    recordSize            INTEGER(1..65535),
    noOfRecords           INTEGER(0..65535),
    records               SET SIZE(noOfRecords) OF
                        VuTimeAdjustmentGNSSRecord
}
```

recordType указывает на тип записи (VuTimeAdjustmentGNSSRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuTimeAdjustmentGNSSRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей о корректировке времени по ГНСС.

2.232. VuTimeAdjustmentRecord

Информация, записанная в бортовом устройстве, которая относится к корректировке времени, не связанной с регулярной калибровкой (приложение 1В, требование 101 и приложение 1С, требования 124 и 125).

Первое поколение:

```
VuTimeAdjustmentRecord ::= SEQUENCE {
    oldTimeValue          TimeReal,
    newTimeValue          TimeReal,
    workshopName          Name,
    workshopAddress       Address,
    workshopCardNumber    FullCardNumber
}
```

oldTimeValue, newTimeValue – старые и новые значения даты и времени.

workshopName, workshopAddress – название и адрес мастерской.

workshopCardNumber – идентификатор карточки мастерской, используемой для корректировки времени.

Второе поколение:

```

VuTimeAdjustmentRecord ::= SEQUENCE {
    oldTimeValue          TimeReal,
    newTimeValue          TimeReal,
    workshopName          Name,
    workshopAddress       Address,
    workshopCardNumberAndGeneration FullCardNumberAndGeneration
}

```

Вместо `workshopCardNumber` в структуре данных второго поколения используется следующий элемент данных.

workshopCardNumberAndGeneration – идентификатор карточки мастерской, используемой для корректировки времени, и её поколения.

2.233. VuTimeAdjustmentRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к корректировке времени, не связанной с регулярной калибровкой (приложение 1С, требования 124 и 125).

```

VuTimeAdjustmentRecordArray ::= SEQUENCE {
    recordType            RecordType,
    recordSize            INTEGER(1..65535),
    noOfRecords           INTEGER(0..65535),
    records                SET SIZE(noOfRecords) OF
                          VuTimeAdjustmentRecord
}

```

recordType указывает на тип записи (`VuTimeAdjustmentRecord`). **Присвоение значения:** см. `RecordType`

recordSize – размер `VuTimeAdjustmentRecord` в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей о корректировке времени.

2.234. WorkshopCardApplicationIdentification

Информация, записанная на карточке мастерской, которая относится к идентификации приложения карточки (приложение 1С, требования 307 и 330).

Первое поколение:

```
WorkshopCardApplicationIdentification ::= SEQUENCE {
    typeOfTachographCardId      EquipmentType,
    cardStructureVersion         CardStructureVersion,
    noOfEventsPerType            NoOfEventsPerType,
    noOfFaultsPerType           NoOfFaultsPerType,
    activityStructureLength      CardActivityLengthRange,
    noOfCardVehicleRecords      NoOfCardVehicleRecords,
    noOfCardPlaceRecords        NoOfCardPlaceRecords,
    noOfCalibrationRecords      NoOfCalibrationRecords
}
```

typeOfTachographCardId уточняет реализованный тип карточки.

cardStructureVersion уточняет версию структуры, реализованной на карточке.

noOfEventsPerType – число событий по типу события, которое может быть записано на карточку.

noOfFaultsPerType – число неисправностей по типу неисправности, которое может быть записано на карточку.

activityStructureLength – число байтов, которые могут быть использованы для хранения записей, относящихся к виду деятельности.

noOfCardVehicleRecords – число записей о транспортном средстве, которые могут храниться на карточке.

noOfCardPlaceRecords – число записей о местах, которые могут храниться на карточке.

noOfCalibrationRecords – число записей о калибровке, которые могут храниться на карточке.

Второе поколение:

```
WorkshopCardApplicationIdentification ::= SEQUENCE {
    typeOfTachographCardId      EquipmentType,
    cardStructureVersion         CardStructureVersion,
    noOfEventsPerType            NoOfEventsPerType,
    noOfFaultsPerType           NoOfFaultsPerType,
    activityStructureLength      CardActivityLengthRange,
    noOfCardVehicleRecords      NoOfCardVehicleRecords,
    noOfCardPlaceRecords        NoOfCardPlaceRecords,
    noOfCalibrationRecords      NoOfCalibrationRecords,
    noOfGNSSCDRecords           NoOfGNSSCDRecords,
    noOfSpecificConditionRecords NoOfSpecificConditionRecords
}
```

В дополнение к первому поколению, используются следующие элементы данных:

noOfGNSSCDRecords – число записей о непрерывном времени управления по ГНСС, которые могут храниться на карточке.

noOfSpecificConditionRecords – число записей об особых условиях, которые могут храниться на карточке.

2.235. WorkshopCardCalibrationData

Информация, записанная на карточке мастерской, которая относится к действиям мастерской, проводимым с карточкой (приложение 1С, требования 314, 316, 337 и 339).

```
WorkshopCardCalibrationData ::= SEQUENCE {
    calibrationTotalNumber      INTEGER(0 .. 216-1),
    calibrationPointerNewestRecord  INTEGER(0 .. NoOfCalibrationRecords-1),
    calibrationRecords          SET SIZE(NoOfCalibrationRecords) OF
                                WorkshopCardCalibrationRecord
}
```

calibrationTotalNumber – общее число калибровок, произведённых с карточкой.

calibrationPointerNewestRecord – индекс последней обновлённой записи данных о калибровке.

Присвоение значения: Число, соответствующее числовому показателю записи данных о калибровке, которое начинается с 0 в случае первой регистрации записей, касающихся калибровки, в структуре.

calibrationRecords – массив данных с записями, содержащими данные о калибровке и/или корректировке времени.

2.236. WorkshopCardCalibrationRecord

Информация, записанная на карточке мастерской, которая относится к калибровке, проводимой с карточкой (приложение 1С, требования 314 и 337).

Первое поколение:

```
WorkshopCardCalibrationRecord ::= SEQUENCE {
    calibrationPurpose          CalibrationPurpose,
    vehicleIdentificationNumber VehicleIdentificationNumber,
    vehicleRegistration         VehicleRegistrationIdentification,
    wVehicleCharacteristicConstant W-VehicleCharacteristicConstant,
    kConstantOfRecordingEquipment K-ConstantOfRecordingEquipment,
    lTyreCircumference         L-TyreCircumference,
    tyreSize                   TyreSize,
    authorisedSpeed             SpeedAuthorised,
    oldOdometerValue           OdometerShort,
    newOdometerValue           OdometerShort,
    oldTimeValue               TimeReal,
    newTimeValue               TimeReal,
    nextCalibrationDate        TimeReal,
    vuPartNumber               VuPartNumber,
    vuSerialNumber              VuSerialNumber,
    sensorSerialNumber          SensorSerialNumber
}
```

calibrationPurpose – цель калибровки.

vehicleIdentificationNumber – идентификационный номер транспортного средства (VIN).

vehicleRegistration содержит VRN и государство-член регистрации.

wVehicleCharacteristicConstant – характеристический коэффициент транспортного средства.

kConstantOfRecordingEquipment – постоянная величина записывающего оборудования.

lTyreCircumference – фактическая окружность шин колёс.

tyreSize – обозначение габаритов шин транспортного средства..

authorisedSpeed – максимальная допустимая скорость транспортного средства.

oldOdometerValue, newOdometerValue – старые и новые показания одометра.

oldTimeValue, newTimeValue – старые и новые значения даты и времени.

nextCalibrationDate – дата следующей калибровки типа, указанного в CalibrationPurpose, которую должен провести компетентный инспекционный орган.

vuPartNumber, **vuSerialNumber** и **sensorSerialNumber** – элементы данных для идентификации записывающего оборудования.

Второе поколение:

```
WorkshopCardCalibrationRecord ::= SEQUENCE {
    calibrationPurpose           CalibrationPurpose,
    vehicleIdentificationNumber  VehicleIdentificationNumber,
    vehicleRegistration          VehicleRegistrationIdentification,
    wVehicleCharacteristicConstant W-VehicleCharacteristicConstant,
    kConstantOfRecordingEquipment K-ConstantOfRecordingEquipment,
    lTyreCircumference          L-TyreCircumference,
    tyreSize                     TyreSize,
    authorisedSpeed              SpeedAuthorised,
    oldOdometerValue             OdometerShort,
    newOdometerValue             OdometerShort,
    oldTimeValue                 TimeReal,
    newTimeValue                 TimeReal,
    nextCalibrationDate          TimeReal,
    vuPartNumber                 VuPartNumber,
    vuSerialNumber               VuSerialNumber,
    sensorSerialNumber           SensorSerialNumber,
    sensorGNSSSerialNumber       SensorGNSSSerialNumber,
    rcmSerialNumber              RemoteCommunicationModuleSerialNumber,
    sealDataCard                 SealDataCard
}
```

В дополнение к первому поколению, используются следующие элементы данных:

sensorGNSSSerialNumber позволяет идентифицировать внешнее устройство ГНСС.

rcmSerialNumber позволяет идентифицировать модуль удалённой связи.

sealDataCard предоставляет информацию о пломбах, наложенных на различные компоненты транспортного средства.

2.237. WorkshopCardHolderIdentification

Информация, записанная на карточке мастерской, которая относится к идентификации владельца карточки (приложение 1С, требования 311 и 334).

```
WorkshopCardHolderIdentification ::= SEQUENCE {
    workshopName                Name,
    workshopAddress              Address,
    cardHolderName               HolderName,
    cardHolderPreferredLanguage  Language
}
```

workshopName – название мастерской владельца карточки.

workshopAddress – адрес мастерской владельца карточки.

cardHolderName – фамилия и имя (имена) владельца (например, фамилия и имя механика).

cardHolderPreferredLanguage – предпочитаемый язык владельца карточки.

2.238. WorkshopCardPIN

Персональный идентификационный номер карточки мастерской (приложение 1С, требования 309 и 332).

```
WorkshopCardPIN ::= IA5String(SIZE(8))
```

Присвоение значения: Известный ПИН-код владельца карточки, за которым следует серия байтов 'FF' (до восьми байтов).

2.239. W-VehicleCharacteristicConstant

Характеристический коэффициент транспортного средства (определение k).

W-VehicleCharacteristicConstant ::= INTEGER(0..2¹⁶-1)

Присвоение значения: Импульсы на километр в рабочем диапазоне от 0 до 64 255 имп./км.

2.240. VuPowerSupplyInterruptionRecord

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к событиям прекращения электропитания (приложение 1С, требование 117).

```
VuPowerSupplyInterruptionRecord ::= SEQUENCE {
    eventType                EventFaultType,
    eventRecordPurpose       EventFaultRecordPurpose,
    eventBeginTime           TimeReal,
    eventEndTime             TimeReal,
    cardNumberAndGenDriverSlotBegin FullCardNumberAndGeneration,
    cardNumberAndGenDriverSlotEnd   FullCardNumberAndGeneration,
    cardNumberAndGenCodriverSlotBegin FullCardNumberAndGeneration,
    cardNumberAndGenCodriverSlotEnd FullCardNumberAndGeneration,
    similarEventsNumber       SimilarEventsNumber
}
```

eventType – тип события.

eventRecordPurpose – цель регистрации данного события.

eventBeginTime – дата и время начала события.

eventEndTime – дата и время завершения события.

cardNumberAndGenDriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство водителя в начале события, и её поколения.

cardNumberAndGenDriverSlotEnd – идентификатор карточки, вставленной в считывающее устройство водителя в конце события, и её поколения.

cardNumberAndGenCodriverSlotBegin – идентификатор карточки, вставленной в считывающее устройство второго водителя в начале события, и её поколения.

cardNumberAndGenCodriverSlotEnd – идентификатор карточки, вставленной в считывающее устройство второго водителя в конце события, и её поколения.

similarEventsNumber – число аналогичных событий за указанный день.

2.241. VuPowerSupplyInterruptionRecordArray

Второе поколение:

Информация, записанная в бортовом устройстве, которая относится к событиям прекращения электропитания (приложение 1С, требование 117).

```

VuPowerSupplyInterruptionRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF
                        VuPowerSupplyInterruptionRecord
}

```

recordType указывает на тип записи (VuPowerSupplyInterruptionRecord). **Присвоение значения:** см. RecordType

recordSize – размер VuPowerSupplyInterruptionRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей о событиях прекращения электропитания.

2.242. VuSensorExternalGNSSCoupledRecordArray

Второе поколение:

Совокупность SensorExternalGNSSCoupledRecord и метаданные, используемые в протоколе загрузки.

```

VuSensorExternalGNSSCoupledRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF
                        SensorExternalGNSSCoupledRecord
}

```

recordType указывает на тип записи (SensorExternalGNSSCoupledRecord). **Присвоение значения:** см. RecordType

recordSize – размер SensorExternalGNSSCoupledRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей SensorExternalGNSSCoupledRecord.

2.243. VuSensorPairedRecordArray

Второе поколение:

Совокупность SensorPairedRecord и метаданные, используемые в протоколе загрузки.

```

VuSensorPairedRecordArray ::= SEQUENCE {
    recordType          RecordType,
    recordSize          INTEGER(1..65535),
    noOfRecords         INTEGER(0..65535),
    records              SET SIZE(noOfRecords) OF
                        SensorPairedRecord
}

```

recordType указывает на тип записи (SensorPairedRecord). **Присвоение значения:** см. RecordType

recordSize – размер SensorPairedRecord в байтах.

noOfRecords – число записей в массиве.

records – совокупность записей о соединениях датчика.

3. Определения диапазонов значений и размеров

Определение значений переменных, используемых для определений, содержащихся в пункте 2.

TimeRealRange ::= $2^{32}-1$

4. Наборы символов

В строках IA5 используются знаки ASCII, определённые в стандарте ISO/IEC 8824-1. Для удобства и простоты присвоенные значения приводятся ниже. В случае разночтений вместо этой информационной записи следует использовать стандарт ISO/IEC 8824-1.

! " # \$ % & ' () * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\] ^ _
` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~ -

Стандартный набор символов	Кодовая страница (десятичная система)
Другие строки символов (Address, Name, VehicleRegistrationNumber) дополнительно задействуют символы из спектра десятичных знаков 161 – 255 следующих 8-битных стандартных наборов символов, указываемых по номеру кодовой страницы:	
ISO/IEC 8859-1 латиница-1 Западная Европа	1
ISO/IEC 8859-2 латиница-2 Центральная Европа	2
ISO/IEC 8859-3 латиница-3 Южная Европа	3
ISO/IEC 8859-5 латиница/кириллица	5
ISO/IEC 8859-7 латиница/греческий алфавит	7
ISO/IEC 8859-9 латиница/турецкий алфавит	9
ISO/IEC 8859-13 латиница-7 Балтика	13
ISO/IEC 8859-15 латиница-9	15
ISO/IEC 8859-16 латиница-10 Восточная Европа	16
KOI8-R латиница/кириллица	80
KOI8-U латиница/кириллица	85

5. Кодировка

В случае кодирования по правилам кодирования ASN.1 все определённые типы данных кодируются в соответствии со стандартом ISO/IEC 8825-2 (согласованный вариант).

6. Идентификаторы объектов и приложений

6.1. Идентификаторы объектов

Идентификаторы объектов (OID), указанные в настоящей главе, касаются только второго поколения. Эти OIDs представлены в TR-03110-3 и повторяются здесь для полноты. OID представлены в поддереве bsid:

```
bsi-de OBJECT IDENTIFIER ::= {
    itu-t(0) identified-organization(4) etsi(0)
    reserved(127) etsi-identified-organization(0) 7
}
```

Идентификаторы протокола аутентификации БУ

```
id-TA OBJECT IDENTIFIER ::= {bsi-de protocols(2) smartcard(2) 2}
id-TA-ECDSA OBJECT IDENTIFIER ::= {id-TA 2}
id-TA-ECDSA-SHA-256 OBJECT IDENTIFIER ::= {id-TA-ECDSA 3}
id-TA-ECDSA-SHA-384 OBJECT IDENTIFIER ::= {id-TA-ECDSA 4}
id-TA-ECDSA-SHA-512 OBJECT IDENTIFIER ::= {id-TA-ECDSA 5}
```

Пример: Допустим, аутентификация БУ должна проходить при помощи SHA-384, и в подобных случаях идентификатор объекта, который следует применить, – (в нотации ASN.1) bsi-de protocols(2) smartcard(2) 2 2 4. Значение данного идентификатора объекта в точечной нотации – 0.4.0.127.0.7.2.2.2.2.4.

	Точечная нотация	Байтовая нотация
id-TA-ECDSA-SHA-256	0.4.0.127.0.7.2.2.2.2.3	'04 00 7F 00 07 02 02 02 03'
id-TA-ECDSA-SHA-384	0.4.0.127.0.7.2.2.2.2.4	'04 00 7F 00 07 02 02 02 04'
id-TA-ECDSA-SHA-512	0.4.0.127.0.7.2.2.2.2.5	'04 00 7F 00 07 02 02 02 05'

Идентификаторы протокола аутентификации микросхемы

```
id-CA OBJECT IDENTIFIER ::= {bsi-de protocols(2) smartcard(2) 3}
id-CA-ECDH OBJECT IDENTIFIER ::= {id-CA 2}
id-CA-ECDH-AES-CBC-CMAC-128 OBJECT IDENTIFIER ::= {id-CA-ECDH 2}
id-CA-ECDH-AES-CBC-CMAC-192 OBJECT IDENTIFIER ::= {id-CA-ECDH 3}
id-CA-ECDH-AES-CBC-CMAC-256 OBJECT IDENTIFIER ::= {id-CA-ECDH 4}
```

Пример: Допустим, аутентификация микросхемы происходит при помощи алгоритма ECDH, и в результате получается ключ сеанса AES длиной 128 бит. Данный ключ сеанса впоследствии используется в рабочем режиме CBC для обеспечения конфиденциальности данных и с алгоритмом CMAC для обеспечения подлинности данных. Таким образом, идентификатор объекта, который следует применить, – (в нотации ASN.1) bsi-de protocols(2) smartcard(2) 3 2 2. Значение данного идентификатора объекта в точечной нотации – 0.4.0.127.0.7.2.2.3.2.2.

	Точечная нотация	Байтовая нотация
id-CA-ECDH-AES-CBC-CMAC-128	0.4.0.127.0.7.2.2.3.2.2	'04 00 7F 00 07 02 02 03 02 02'
id-CA-ECDH-AES-CBC-CMAC-192	0.4.0.127.0.7.2.2.3.2.3	'04 00 7F 00 07 02 02 03 02 03'
id-CA-ECDH-AES-CBC-CMAC-256	0.4.0.127.0.7.2.2.3.2.4	'04 00 7F 00 07 02 02 03 02 04'

6.2. Идентификаторы приложения

Второе поколение:

Идентификатор приложения (AID) для внешнего устройства ГНСС (второго поколения) происходит из 'FF 44 54 45 47 4D'. Это собственный AID согласно ISO/IEC 7816-4.

Примечание: Последние 5 байтов кодируют DTEGM для внешнего устройства ГНСС «умных» тахографов.

Идентификатор приложения для приложения карточки тахографа второго поколения происходит из 'FF 53 4D 52 44 54'. Это собственный AID согласно ISO/IEC 7816-4.