

ВСТРОЕННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
КОНТРОЛЛЕРА ЗАРЯДА CHAdEMO

Описание функциональных характеристик



ООО «Силовая электроника»

СОДЕРЖАНИЕ

1. Общие сведения.....	2
2. Описание программного интерфейса контроллера CHAdEMO при взаимодействии по RPC.....	2
3. Процедуры программного интерфейса контроллера.....	3
3.1 Передача текущего статуса зарядной станции	3
3.2 Запуск и остановка зарядной сессии	4
3.3 Установка заданных выходных параметров зарядной станции.....	4
3.4 Передача текущего статуса зарядной сессии	5
3.5 Перезагрузка встроенного программного обеспечения контроллера заряда.....	8
3.6 Передача версии программного обеспечения контроллера заряда.....	8
4. Требования к аппаратному обеспечению	8

1. Общие сведения

Встроенное программное обеспечение (ПО) контроллера заряда CHAdeMO предназначено для управления зарядной сессией в соответствии со стандартом CHAdeMO 1.2 в составе электростанции.

Встроенное программное обеспечение контроллера заряда CHAdeMO обеспечивает:

- прием параметров зарядной сессии и управляющих команд по интерфейсу Ethernet;
- выдачу информации о ходе и параметрах текущей зарядной сессии по интерфейсу Ethernet;
- цифровой обмен данными между электромобилем и зарядной станцией по шине CAN 2.0B J1939;
- управление и контроль состояния внешнего электромеханического устройства блокировки разъема зарядного кабеля посредством сигналов Permission, CSS1, CSS2, Lock Solenoid, Lock Detect;
- контроль температуры контактов зарядного разъема по электрическому сопротивлению встроенных в него термодатчиков;
- управления светодиодной индикацией состояния зарядной сессии.

2. Описание программного интерфейса контроллера CHAdeMO при взаимодействии по RPC

Программный интерфейс взаимодействия контроллера заряда CHAdeMO с контроллером ЭЗС основан на использовании удаленного вызова процедур (RPC) по сетевому протоколу TCP/IP v4. Используемый формат сообщений msgpack-rpc. Для данного протокола взаимодействия существуют библиотеки с открытым исходным кодом под разные платформы и языки программирования.

Контроллер сконфигурирован на получение параметров (IP адрес, маска подсети, шлюз), необходимых для работы в сети TCP/IP, по протоколу DHCP при загрузке. Также по протоколу DHCP контроллеру может быть передан адрес NTP сервера для синхронизации времени.

При отсутствии в сети DHCP сервера контроллеру присваивается статический IP адрес 192.168.3.222/24.

Контроллер реализует одновременно сервер RPC (для приема команд от контроллера зарядной станции) и клиент RPC (для передачи команд на контроллер зарядной станции).

Обмен информацией между управляющим контроллером и контроллером заряда CHAdeMO осуществляется по схеме Master-Slave, при этом контроллер CHAdeMO выступает в роли Slave. Для инициализации соединения клиент RPC управляющего

контроллера подключается к серверу RPC контроллера CHAdeMO на порт 18000 и выполняет метод **«rpcConnectRequest»**:

```
string rpcConnectRequest(string interfacedId, string remoteAddress, int remotePort,  
                           time_t connectionTimeoutMsec, time_t pingPeriodMsec,  
                           uint32_t pingCheckCount);
```

interfacedId – идентификатор интерфейса RPC, должен быть равен IID_SECC_CHADEMO_1.0

remoteAddress – собственный IP адрес управляющего контроллера;

remotePort – порт сервера RPC управляющего контроллера;

connectionTimeoutMsec – таймаут соединения TCP, мсек;

pingPeriodMsec – требуемый период передачи пингов, мсек;

pingCheckCount – количество пропущенных пингов, после которого соединение считается разорванным.

После получения вызова **«rpcConnectRequest»** клиент RPC контроллера CHAdeMO выполняет обратное подключение к серверу RPC управляющего контроллера по указанному IP адресу и порту и начинает с периодичностью pingPeriodMsec вызывать метод **«rpcPing»** для поддержания соединения:

```
void rpcPing(int selfConnectionInputState, int selfConnectionOutputState);
```

selfConnectionInputState – принимаются ли пинги с удаленной стороны (1 – пингов нет, 2 – пинги есть);

selfConnectionOutputState – отправляются ли пинги к удаленной стороне (1 – ошибка отправки пингов, 2 – пинги отправляются).

Клиент RPC управляющего контроллера после установки соединения также должен с периодичностью pingPeriodMsec осуществлять вызов метода **«rpcPing»** для поддержания соединения и передавать актуальные значения selfConnectionInputState, selfConnectionOutputState.

Если сервер RPC контроллера CHAdeMO после установки соединения не получает пинги в течение времени pingPeriodMsec * pingCheckCount или происходит таймаут соединения TCP, то соединение клиента RPC контроллера CHAdeMO к серверу RPC управляющего контроллера разрывается и устанавливается заново.

3. Процедуры программного интерфейса контроллера

3.1 Передача текущего статуса зарядной станции

После установки RPC соединения, а затем при изменении одного из параметров или периодически контроллер зарядной станции выполняет метод **«SET_INVERTOR_STATE»** для передачи информации о текущем состоянии зарядной станции.

```
void SET_INVERTOR_STATE (uint32_t mode, uint32_t errors, uint32_t warnings,
                        float maximumPowerLimit, float maximumVoltageLimit,
                        float maximumCurrentLimit, float minimumVoltageLimit,
                        float minimumCurrentLimit, float targetVoltage,
                        float targetCurrent, float presentVoltage,
                        float presentCurrent, float reserved1,
                        float reserved2, bool reserved3);
```

SET_INVERTOR_STATE			в контроллер
Аргумент	Тип	Описание	Значение
mode	uint32_t	текущий режим работы силовых блоков	0x00 – авария 0x01 – режим ожидания 0x02 – режим заряда 0x03 – проверка изоляции 0x0F – выключено
errors	uint32_t	ошибки зарядной станции и контроля изоляции	0x00 – нет ошибок 0x01 – ошибка силовой части зарядной станции 0x02 – ошибка при контроле изоляции
status	uint32_t	статус контроля изоляции	0x00 – проверка изоляции завершена 0x08 – идет проверка изоляции
maximumPowerLimit	float	максимальная выходная мощность станции	Вт
maximumVoltageLimit	float	максимальное выходное напряжение станции	В
maximumCurrentLimit	float	максимальный выходной ток станции	А
minimumVoltageLimit	float	минимальное выходное напряжение станции	В
minimumCurrentLimit	float	минимальный выходной ток станции	А
targetVoltage	float	установленное выходное напряжение	В
targetCurrent	float	установленный выходной ток	А
presentVoltage	float	текущее выходное напряжение	В
presentCurrent	float	текущий выходной ток	А
reserved1	float	зарезервировано	должно быть = 0
reserved2	float	зарезервировано	должно быть = 0
reserved3	bool	зарезервировано	должно быть = false

3.2 Запуск и остановка зарядной сессии

После установки RPC соединения и подключения зарядного кабеля к электромобилю для старта зарядной сессии контроллер зарядной станции должен выполнить метод «**AUTHORIZE**».

```
void AUTHORIZE ();
```

Для остановки начатой зарядной сессии контроллер зарядной станции должен выполнить метод «**USER_STOP**».

```
void USER_STOP ();
```

3.3 Установка заданных выходных параметров зарядной станции

После установки RPC соединения для управления силовыми преобразователями зарядной станции контроллер выполняет метод «**SET_INVERTOR_SET**».

```
void SET_INVERTOR_SET (uint32_t mode, float reserved1, float reserved2,
float reserved3, float reserved4, float reserved5,
float targetVoltage, float targetCurrent);
```

SET_INVERTOR_SET			из контроллера
Аргумент	Тип	Описание	Значение
mode	uint32_t	заданный режим работы силовых блоков	0x01 – включить режим ожидания 0x02 – включить режим заряда 0x03 – включить проверку изоляции 0x0F – выключить
reserved1	float	зарезервировано	
reserved2	float	зарезервировано	
reserved3	float	зарезервировано	
reserved4	float	зарезервировано	
reserved5	float	зарезервировано	
targetVoltage	float	заданное выходное напряжение	V
targetCurrent	float	заданный выходной ток	A

3.4 Передача текущего статуса зарядной сессии

При изменении одного из параметров зарядной сессии контроллер выполняет метод «**SET_CHADEMO**» для передачи информации о текущем состоянии зарядной сессии и информации от электромобиля:

```
void SET_CHADEMO (uint32_t state, uint32_t protocol, bool reserved1, bool reserved2,
bool noInputCAN, float evMinimumCurrent,
float evMaximumBatteryVoltage, float chargedRateReference,
float totalCapacityBattery, float evTargetBatteryVoltage,
float evChargingCurrentRequest, float availableOutputCurrent,
float thresholdVoltage, unsigned int evStateOfCharge,
float evMaximumChargingTime, float eVEstimatedTime,
float reserved3, float remainingChargingTime,
float chargingTime, uint32_t reserved4,
bool batteryOvervoltage, bool batteryUndervoltage,
bool batteryCurrentDeviation, bool highBatteryTemperature,
bool batteryVoltageDeviation, bool reservedFault,
bool vehicleChargingEnabled, bool vehicleShiftPosition,
bool chargingSystemErrorStatus, bool vehicleStatus,
bool stopBeforeCharging, bool reservedStatus, bool chargerStatus,
bool chargerError, bool energizingState, bool batteryIncompatibility,
bool chargingSystemErrorStFlt, bool chargingStopControl,
bool controllerError, bool A_1, bool A_2, bool A_3, bool A_4, bool
A_5, bool A_6, bool A_7, bool A_8);
```

SET_CHADEMO		из контроллера	
Аргумент	Тип	Описание	Значение
state	uint32_t	Стадия заряда	<p>cs_DISCONNECTED = 0, Unconnected with vehicle</p> <p>cs_B_start = 16, Connected with vehicle</p> <p>cs_C1 = 17, From [Switch(d1) = ON] to Reception of Vehicle data (#100, #101, #102)</p> <p>cs_C2 = 18, From Reception of Vehicle data to [opto-coupler (j) = ON and "vehicle charging enabled (H'102.5.0) = 1]</p> <p>cs_C3 = 19, From [opto-coupler (j) = ON to "vehicle charging enabled (H'102.5.0) = 1]</p> <p>cs_D1 = 32, From [opto-coupler (j) = ON and "vehicle charging enabled"(H'102.5.0) = 1] to [Switch(d2) = ON]</p> <p>cs_D2 = 33, From [Switch(d2) = ON] to [Uout >= 50 V and vehicle status (H'102.5.3) = 0]</p> <p>cs_D3 = 34, From [Uout >= 50 V and vehicle status (H'102.5.3) = 0] to [charger status (H'109.5.0) = 1 and charging stop control (H'109.5.5) = 0]</p> <p>cs_E = 64, From [charger status (H'109.5.0) = 1 and charging stop control (H'109.5.5) = 0] via [charging] to [transmission / reception of stop instruction]</p> <p>cs_F1 = 65, From [transmission / reception of stop instruction] to [current in the output circuit is less than or equal to 5A and charger status (H'109.5.0) = 0]</p> <p>cs_F2 = 66, From [transmission / reception of stop instruction] to [current in the output circuit is less than or equal to 5A and charger status (H'109.5.0) = 0]</p> <p>cs_B002F = 67, From [transmission / reception of stop instruction] to [current in the output circuit is less than or equal to 5A and charger status (H'109.5.0) = 0]</p> <p>cs_G = 80, From [current in the output circuit is less than or equal to 5A and charger status (H'109.5.0) = 0] to [vehicle status (H'102.5.3) = 1]</p> <p>cs_B4B5F3 = 96,</p> <p>cs_H1 = 97, From [vehicle status (H'102.5.3) = 1] and [Switch(d1) = OFF] to [Switch(d2) = OFF] cs_H2 = 98, Delay 50 msec before H3</p> <p>cs_H3 = 99, From [Switch(d2) = OFF] to [voltage in the output circuit is less than or equal to 10V and energizing state (H'109.5.2) = 0]</p> <p>cs_B002H1 = 100, From [vehicle status (H'102.5.3) = 1] and [Switch(d1) = OFF] to [Switch(d2) = OFF]</p> <p>cs_B002H2 = 101, From [Switch(d2) = OFF] to [voltage in the output circuit is less than or equal to 10V and energizing state (H'109.5.2) = 0]</p> <p>cs_I = 102, From [voltage in the output circuit is less than or equal to 10V and energizing state (H'109.5.2) = 0] to [End of CAN data transmission]</p> <p>cs_SESSION_END = 128, [End of CAN data transmission]</p>
protocol	uint32_t	Версия протокола	<p>0x00: CHAdeMO specification 0.9 and earlier</p> <p>0x01: 0.9 and 0.9.1,</p> <p>0x02: 1.0.0, 1.0.1, 1.1, 1.2</p>
reserved1	bool	Не использ.	
reserved2	bool	Не использ.	
noInputCAN	bool	Наличие кадров CAN	<p>false = производится обмен кадрами с электромобилем по CAN</p> <p>true = входные кадры отсутствуют в течение > 200 мсек</p>

eVMinimumCurrent	float	Минимально допустимый зарядный ток	A
eVMaximumBatteryVoltage	float	Максимальное напряжение на АКБ	B
chargedRateReference	float	Максимальный уровень заряда АКБ	%
totalCapacityBattery	float	Емкость АКБ	Вт*час
evTargetBatteryVoltage	float	Запрашиваемое напряжение	B
evChargingCurrentRequest	float	Запрашиваемый ток	A
availableOutputCurrent	float	Максимальный ток зарядной станции	A
thresholdVoltage	float	Пороговое значение напряжения зарядной станции	B
evStateOfCharge	unsigned int	Текущий уровень заряда АКБ	%
evMaximumChargingTime	float	Максимальное время заряда	сек
evEstimatedTime	float	Расчетное время заряда	сек
reserved3	float	Не используется	
remainingChargingTime	float	Оставшееся время заряда	сек
chargingTime	float	Текущая длительность зарядной сессии	мсек
reserved4	uint32_t	Не используется	
batteryOvervoltage	bool	Флаги состояния электромобиля и зарядной станции	В соответствии с протоколом CHAdeMO
batteryUndervoltage	bool		
batteryCurrentDeviation	bool		
highBatteryTemperature	bool		
batteryVoltageDeviation	bool		
reservedFault	bool		
vehicleChargingEnabled	bool		
vehicleShiftPosition	bool		
chargingSystemErrorStatus	bool		
vehicleStatus	bool		
stopBeforeCharging	bool		
reservedStatus	bool		
chargerStatus	bool		
chargerError	bool		
energizingState	bool		
batteryIncompatibility	bool		
chargingSystemErrorStFlt	bool		
chargingStopControl	bool		
controllerError	bool	Неисправность контроллера	false = нет ошибок true = неисправность контроллера
A_1	bool	События зарядной сессии	В соответствии с протоколом CHAdeMO
A_2	bool		
A_3	bool		
A_4	bool		
A_5	bool		
A_6	bool		
A_7	bool		
A_8	bool		

3.5 Перезагрузка встроенного программного обеспечения контроллера заряда

После установки RPC соединения при необходимости перезагрузить контроллер заряда контроллер зарядной станции должен выполнить метод «**RESET**».

```
void RESET ();
```

3.6 Передача версии программного обеспечения контроллера заряда

После установки RPC соединения контроллер заряда передает версию своего программного обеспечения с помощью вызова метода «**SETVERSION**».

```
void SETVERSION (string version);
```

SETVERSION			из контроллера
Аргумент	Тип	Описание	Значение
version	string	текущая версия ПО контроллера	

4. Требования к аппаратному обеспечению

Минимальные требования к аппаратному обеспечению для функционирования встроенного программного обеспечения контроллера CHAdeMO:

- процессорное ядро ARM Cortex-A7;
- тактовая частота не менее 500 МГц;
- объем Flash памяти – 256 МБ;
- объем ОЗУ – 256 МБ.