# Batcharging Protocol Data Unit

# 一.通信帧格式

通信帧的格式如下表1-1所示：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 协议头 | 协议版本号 | 加密子 | 总包长度 | 指令ID | 序列号 | 负载 | 校验码 |
| bMagic | bVersion | bKey | nLength | nCmdId | nSeq | payload | CRC16 |
| 0x55AA | 0x01 |  |  |  | 自增 |  |  |
| 2B | 1B | 1B | 2B | 2B | 2B |  | 2B |

# 二.通信指令

|  |  |  |  |
| --- | --- | --- | --- |
| No | 操作码 | 说明 | 指令方向 |
|  | 0x01, 0x00 | EVSE心跳请求 | EVSE → CSMS |
|  | 0Xa1, 0x00 | Server心跳应答 | CSMS→ EVSE |
|  | 0x01, 0x01 | EVSE登录请求 | EVSE→ CSMS |
|  | 0xA1, 0x01 | EVSE登录应答 | CSMS→ EVSE |
|  | 0x01, 0x02 | EVSE本地授权请求 | EVSE→ CSMS |
|  | 0xA1, 0x02 | EVSE本地授权应答 | CSMS→ EVSE |
|  | 0x02, 0x01 | 远程授权充电 | CSMS→ EVSE |
|  | 0xA2, 0x01 | EVSE远程授权充电应答 | EVSE→ CSMS |
|  | 0x01, 0x03 | 事务事件 | EVSE→ CSMS |
|  | 0xA1, 0x03 | 事务事件应答 | CSMS→ EVSE |
|  | 0x01, 0x04 | 工作数据上报 | EVSE→ CSMS |
|  | 0xA1, 0x04 | 工作数据上报应答 | CSMS→ EVSE |
|  | 0x02, 0x02 | 后台停止充电 | CSMS→ EVSE |
|  | 0xA2, 0x02 | 后台停止充电应答 | EVSE→ CSMS |
|  | 0x01, 0x05 | 事件上报 | EVSE→ CSMS |
|  | 0xA1, 0x05 | 事件上报应答 | CSMS→ EVSE |
|  | 0x01, 0x06 | 插头状态上报 | EVSE→ CSMS |
|  | 0xA1, 0x06 | 插头状态上报应答 | CSMS→ EVSE |
|  | 0x02, 0x03 | 更新充电金额/时间 | CSMS→ EVSE |
|  | 0xA2, 0x03 | 更新充电金额/时间应答 | EVSE→ CSMS |
|  | 0x02, 0x04 | 设置组件变量 | CSMS→ EVSE |
|  | 0xA2, 0x04 | 设置组件变量应答 | EVSE→ CSMS |
|  | 0x02, 0x05 | 获取组件变量 | CSMS→ EVSE |
|  | 0xA2, 0x05 | 获取组件变量应答 | EVSE→ CSMS |

# 三.协议数据单元

## 3.1 心跳指令：

*EVSE----🡪CSMS*

message beatHeartReq{

optional int32 csq; //信号强度

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| csq | 信号强度 |  |

*CSMS ----🡪EVSE*

message beatHeartResp{

required bytes timestamp = 1;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| timestamp | 时间(BCD) | √ |

## 3.2 登录指令：

*EVSE----🡪CSMS*

syntax = "proto2";

message bootNotificationReq{

enum bootReasonEnumType{

ApplicationReset = 0;

FirmWareUpdate = 1;

LocalReset = 2;

PowerUp = 3;

RemoteReset = 4;

Watchdog = 5;

CommunicationTimeout = 6;

Unknown = 7;

};

message modemType{

optional string iccid = 1;

optional string imsi = 2;

}

required bootReasonEnumType reason = 1;

required string serialNumber = 2;

required string key = 3;

optional string mode = 4;

optionalstring vendorName = 5;

optionalstring firmwareVersion = 6;

required uint32 pcbID = 7;

optional modemType modem = 8;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| Reason | 登陆原因 | √ |
| serialNumber | Serial number | √ |
| Key | 登录秘钥 | √ |
| Mode | 设备工作的模式 |  |
| vendorName | 卖主名 |  |
| firmwareVersion | 固件版本 |  |
| pcbId | 固件配置ID | √ |
| Mode | 手机卡标识信息 |  |

*CSMS ----🡪EVSE*

message bootNotificationResp{

enum registrationStatusEnumType{

Accepted = 0;

Pending = 1;

Rejected = 2;

};

required registrationStatusEnumType status = 1;

required bytes currentTime = 2;

required int32 interval = 3;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| Status | 登录状态 | √ |
| currentTime | 时间 | √ |
| Interval | 心跳间隔（Accepted）  下次登陆请求间隔（Pending） | √ |

## 3.3 本地授权充电：

*EVSE----🡪CSMS*

syntax = "proto2";

import "IDTokenType.proto";

import "IDTokenInfoType.proto";

message authorizeReq{

required IDTokenType idToken = 1;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| idToken | 身份信息 | √ |

*CSMS ----🡪EVSE*

message authorizeResp{

enum certificateStatusEnumType{

Accepted = 0;

SignatureError = 1;

CertificateExpired = 2;

NoCertificateAvailable = 3;

}

required certificateStatusEnumType certificateStatus = 1;

required IDTokenInfoType idTokenInfo = 2;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| certificateStatus | 认证状态 | 对 |
| idTokenInfo | 认证ID的私人数据 | 对 |

## 3.4 远程授权充电:

*EVSE----🡪CSMS*

msyntax = "proto2";

import "IDTokenType.proto";

message startTransactionCmd{

enum chargeModeEnumType{

ChargeByHour = 0;

ChargeByMoney = 1;

ChargeByEnergy = 2;

}

message chargingProfileType{

required chargeModeEnumType chargeMode = 1;

required uint32 value = 2;

optional uint32 powerLimit = 3;

}

required uint32 connectorId = 1;

required IDTokenType IDToken = 2;

required chargingProfileType chargingProfile = 3;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| sn | Serial number | √ |
| connectorId | 插头ID | √ |
| IDToken | 身份信息 | √ |
| chargingProfile | 充电属性 | √ |

*CSMS ----🡪EVSE*

message startTransactionRelay{

enum startTransactionStatusEnumType{

Accepted = 0;

Rejected = 1;

}

enum startTransactionErrorEnumType{

ConnectorUnavailable = 0;

ConnectorFault = 2;

ConnectorInUse = 3;

EVSEUnavailable = 4;

EVSEFault = 5;

};

required startTransactionStatusEnumType status = 1;

optional startTransactionErrorEnumType errorCode = 2;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| status | 远程开始充电状态 | √ |
| errorCode | 错误码 |  |

## 3.5 事务事件:

*EVSE----🡪CSMS*

syntax = "proto2";

import "IDTokenType.proto";

message transactionEventReq{

enum transactionEventEnumType{

Ended = 0;

Started = 1;

Updated = 2;

}

enum chargingStateEnumType{

Charging = 0;

Suspended = 1;

}

enum reasonEnumType{

DeAuthorized = 0;

Local = 1;

EnergyLimitReached = 2;

EVDisconnected = 3;

GroundFault = 4;

OvercurrrendFault = 5;

PowerLoss = 6;

PowerQuality = 7;

Remote = 8;

TimeLimitReached = 9;

Timeout = 10;

SOCLimitReached = 11;

ImmediateReset = 12;

}

message transactionType{

optional chargingStateEnumType chargingState = 1;

optional uint32 timeSpentCharging = 2;

optional reasonEnumType stoppedReason = 3;

optional bytes transactionID = 4;

}

required transactionEventEnumType eventType = 1;

required bytes timestamp = 2;

required transactionType transaction = 3;

optional IDTokenType IDToken = 4;

optional uint32 connectorId = 5;

optional uint32 totalTime = 6;

optional uint32 totalCost = 7;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| eventType | 事件类型(充电开始/结束/更新) | √ |
| timestamp | 时间 | √ |
| transaction | 事务 | √ |
| IDToken | 授权身份ID |  |
| connectorId | 插座号 |  |
| totalTime | 总共时间 |  |
| totalCost | 总共花费 |  |

*CSMS ----🡪EVSE*

message transactionEventResps{

enumtransactionStatusEnumType{

Accepted = 0;

TranssctionProcing = 1;

TransactionIdExist = 2;

}

required transactionStatusEnumType status = 1;

optional TransactionID = 2;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| status | 状态 | √ |
| transactionID | 最新订单ID |  |

## 3.6 工作数据：

*EVSE----🡪CSMS*

syntax = "proto2";

message chargingDataReportReq{

message chargingDataType{

required uint32 connectorId = 1;

optional uint32 cost = 2;

optional uint32 voltage = 3;

optional uint32 current = 4;

optional uint32 costTime = 5;

}

repeated chargingDataType chargingData = 1;

required bytes timestap = 2;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| chargingData | 充电工作数据 | √ |
| timestamp | 时间 | √ |

*CSMS ----🡪EVSE*

message chargingDataReportResp{

}

## 3.7后台停充

*EVSE----🡪CSMS*

message stopTransactionCmd{

required bytes transactionID = 1;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| transactionID | 订单号 | √ |

*CSMS ----🡪EVSE*

message stopTransactionRelay{

enum errorCodeEnumType{

TransactionIDNotFound = 0;

}

enum stopStatusEnumType{

Accepted = 0;

Rejected = 1;

}

required stopStatusEnumType status;

optional errorCodeEnumType errorCode;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| status | 停充状态 | √ |

## 3.8事件上报：

*EVSE----🡪CSMS*

import componentType.proto

import variableType.proto

message notifyEventReq{

required bytes timestamp = 1;

required string actualValue = 2;

required componentType component = 3;

required variableType variable = 4;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| timestamp | 时间 | √ |
| actualValue | 实际值 | √ |
| component | 组件 | √ |
| variable | 变量 | √ |

*CSMS ----🡪EVSE*

message notifyEventResp{

}

## 3.9插头状态上报

*EVSE----🡪CSMS*

syntax = "proto2";

message statusNotificationReq{

enum connectorStatusEnumType{

Available = 0;

Occupied = 1;

Reserved = 2;

Unavailable = 3;

Faulted = 4;

}

message connectorsStatusType{

required uint32 connectorId = 1;

required connectorStatusEnumType connectorStatus = 2;

optional int32 faultCode = 3;

}

required bytes timestamp = 1;

repeated connectorsStatusType connectorsStatus = 2;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| timestamp | 时间 | √ |
| connectorsStatus | 插头状态 | √ |

*CSMS ----🡪EVSE*

message statusNotificationResp{

}

## 3.10更新充电金额/时间：

*EVSE----🡪CSMS*

message costUpdatedCmd{

optional uint32 totalCost = 1;

optional uint32 totalTime = 2;

required bytes transactionID = 3;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| totalCost | 总金额 |  |
| totalTime | 总时间 |  |
| transactionId | 订单号 | √ |

*CSMS ----🡪EVSE*

message costUpdatedReplay{

enum costUpdatedStatusEnumType{

Accepted = 0,

Rejected = 1,

}

required costUpdatedStatusEnumType status = 1;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| status | 状态 | √ |

## 3.11设置组件变量

*EVSE----🡪CSMS*

syntax = "proto2";

import "setVariableDataType.proto";

import "setVariableResultType.proto";

message setVariablesCmd{

repeated setVariableDataType setVariableData = 1;

}

syntax = "proto2";

import "componentType.proto";

import "variableType.proto";

import "attributeEnumType.proto";

message setVariableDataType{

optional attributeEnumType attributeType = 1;

required string attributeValue = 2;

required componentType component = 3;

required variableType variable = 4;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| attributeType | 设置属性类型 |  |
| attributeValue | 设置的属性值 | √ |
| component | 组件 | √ |
| variable | 变量 | √ |

*CSMS ----🡪EVSE*

message setVariablesReply{

repeated setVariableResultType setVariableResult = 1;

}

syntax = "proto2";

import "attributeEnumType.proto";

import "componentType.proto";

import "variableType.proto";

message setVariableResultType{

enum setVariableStatusEnumType{

Accepted = 0;

Rejected = 1;

InvalidValue = 2;

UnknownComponent = 3;

UnknownVariable = 4;

UnknowAttributeType = 5;

OutOfRange = 6;

RebootRequired = 7;

}

required setVariableStatusEnumType attributeStatus = 1;

optional attributeEnumType attributeType = 2;

optional componentType component = 3;

optional variableType variable = 4;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| attributeStatus | 设置结果 | √ |
| component | 组件 |  |
| variable | 变量 |  |

## 3.12获取组件变量

*EVSE----🡪CSMS*

import "getVariableDataType.proto"

message getVariableCmd{

repeated getVariableDataType getVariablesData = 1;

}

syntax = "proto2";

import "attributeEnumType.proto";

import "componentType.proto";

import "variableType.proto";

message getVariableDataType{

optional attributeEnumType attributeType = 1;

required componentType component = 2;

required variableType variable = 3;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| attributeType | 设置的属性类型 |  |
| component | 组件 | √ |
| variable | 变量 | √ |

*CSMS ----🡪EVSE*

*import getVariableResultType.proto*

*message getVariableReply{*

*repeated getVariableResultType getVariables = 1;*

*}*

import "attributeEnumType.proto"

import "componentType.proto"

import "variableType.proto"

message getVariableResultType{

optional attributeEnumType attributeType = 1;

required string attributeValue = 2;

optional componentType component = 3;

optional variableType variable = 4;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| attributeType | 设置的属性类型 |  |
| attributeValue | 属性值 | √ |
| component | 组件 |  |
| variable | 变量 |  |

## 3.13 固件更新

*CSMS ----🡪EVSE*

syntax = "proto2";

message updateFirmwareEventReq{

enum updateEventTypeEnumType{

Started = 0;

Canceled = 1;

Paused = 2;

}

enum updateMethodEnumType{

UpdateByCSMS = 0;

UpdateByFTP = 1;

}

message updateByFTPData{

required string location = 1;

optional int32 retries = 2 [default = 3];

}

message updateParamType{

required updateMethodEnumType updateMethod = 1;

optional string location = 2;

optional int32 retries = 3;

optional int32 maxPackageSize = 4;

}

required updateFirmwareStatusEnumType eventType = 1;

optional updateParamType updateParam = 2;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| eventType | 固件更新事件 | √ |
| updateParam | 参数 |  |

*EVSE----🡪CSMS*

message updateFirmwareEventResp{

enum updateFirmwareStatusEnumType{

Accepted = 0;

Rejected = 1;

AcceptedCanceled = 2;

}

required updateFirmwareStatusEnumType status = 1;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| status | 状态 | √ |

## 3.14固件更新状态上报

syntax = "proto2"

message firmwareStatusNotificationReq{

enum firmwareStatusEnumType{

Downloaded = 0;

DownloadFailed = 1;

Downloading = 2;

DownloadPaused = 3;

Idle = 4;

Installing = 5;

Installed = 6;

InvalidCertificate = 7;

InvalidSignature = 8;

InstallRebooting = 9;

NoFirmware = 10;

}

required firmwareStatusEnumType status = 1;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| status | 状态 | √ |

message firmwareStatusNotificationResp{

}

## 3.15数据传输

这条指令是双向的。

syntax = "proto2"

message dataTransferReq{

required string messageID = 1;

optional bytes data = 2;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| messageID | 传输/请求的消息 | √ |
| data | 传输的数据 |  |

message dataTransferResp{

enum dataTransferStatusEnumType{

Accepted = 0;

Rejected = 1;

UnknownMessageId = 2;

}

required dataTransferStatusEnumType status = 1;

optional bytes data = 2;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| status | 状态 | √ |
| data | 请求的数据 |  |

## 3.16触发消息

syntax = "proto2";

message triggerMessageReq{

enum messageTriggerEnumType{

BootNotification = 0;

FirmwareStatusNotification = 1;

Heartbeat = 2;

StatusNotification = 3;

TransactionEvent = 4;

}

required messageTriggerEnumType requestedMessage = 1;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| requestedMessage | 请求发送的消息 | √ |

message triggerMessageResp{

enum triggerMessageStatusEnumType{

Accepted = 0;

Rejected = 1;

NotImplemented = 2;

}

required triggerMessageStatusEnumType status;

}

|  |  |  |
| --- | --- | --- |
| 变量名 | 描述 | 必要 |
| status | 状态 | √ |

# 四．工作流程

## 4.1本地授权充电

*EVSE --🡪CSMS*

authorizeRequest(idToken = “12345”, sn=”32414231”);

CSMS-🡪EVSE

authorizeResponse(certificateStatus = Accepted, idTokenInfo.balance = 100 );

*EVSE --🡪CSMS*

transactionEventRequest(eventType = Started, timestamp=18/8/26 9:22,

transaction.transactionID = ”188269223241423100001”,

connectorId = 2,

totalTime = 180);

CSMS-🡪EVSE

transactionEventResponse(status = Accepted);

## 4.2 本地停止充电

人为造成的停止充电（拔插头）

*EVSE --🡪CSMS*

transactionEventRequest（eventType = Ended,timestamp=18/8/26 9:30,

transaction.stoppedReason = Local,

transaction.timeSpentCharging =20,

transaction.transactionID = ”3241423100001”）;

CSMS-🡪EVSE

transactionEventResponse(status = Accepted);

## 4.3 远程触发充电

CSMS-🡪EVSE

//

startTransactionCmd(idToken = “12345”,connectorId = 2,

chargingProfile.chargeMode = chargeByHour,

chargingProfile.value = 180,

[chargingProfile.powerLimit = 100W] );

*EVSE --🡪CSMS*

startTransactionRelay(status = Accepted)

*EVSE --🡪CSMS*

transactionEventRequest(eventType = Started, timestamp=18/8/26 9:22,

transaction.transactionID = ”3241423100001”,

connectorId = 2,

totalTime = 180);

CSMS-🡪EVSE

transactionEventResponse(status = Accepted);

## 4.4 远程停止充电

CSMS-🡪EVSE

stopTransactionCmd（transactionID =”3241423100001”）

*EVSE --🡪CSMS*

stopTransactionRelay（status = Accepted）

*EVSE --🡪CSMS*

transactionEventRequest（eventType = Ended,timestamp=18/8/26 9:30,

transaction.stoppedReason = Remote,

transaction.timeSpentCharging =20,

transaction.spendEneragyTotal = 2000W,

transaction.spendCost = 200,

transaction.transactionID = ”3241423100001”）;

CSMS-🡪EVSE

transactionEventResponse(status = Accepted);

## 4.5 固件更新

CSMS-🡪EVSE

updateFirmwareEventRequest（eventType = Started,

updateParam.updateMethod = UpdateByTCP

updateParam.retries = 3, updateParam.maxPackageSize = 400）

*EVSE --🡪CSMS*

updateFirmwareEventResponse(status = Accepted)

*EVSE --🡪CSMS*

firmwareStatusNotificationRequest(status = Downloading)

CSMS-🡪EVSE

firmwareStatusNotificationResponse()

CSMS-🡪EVSE

dataTransferRequest(messageID=firmwareUpdate,

data = xxxxxxxxxxxxxxxxxxxx)

*EVSE --🡪CSMS*

dataTransferResponse(status = Accepted)

.

.

.

*EVSE --🡪CSMS*

firmwareStatusNotificationRequest(status = Downloaded)

CSMS-🡪EVSE

firmwareStatusNotificationResponse()

*EVSE --🡪CSMS*

firmwareStatusNotificationRequest(status = installed)

CSMS-🡪EVSE

firmwareStatusNotificationResponse()

# 五．组件及变量

## 5.1 Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常见变量 | 描述 | 类型 | 监控 | 权限 |
| Active | 是否起作用 | Boolean | false | R |
| Interval[HeartBeat] | 心跳间隔 | Integer | false | RW |
| Manufacturer | 出厂名 | String | false | R |
| Type | 设备类型 | OptionList | false | RW |
| SerialNumber | 序列号 | String | false | RW |
| VersionNumber | 软件版本号 | Integer | false | R |
| VersionData | 软件发布日期 | DateTime | false | R |
| SQCode | 二维码 | String | True | RW |

## 5.2 Connector

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常见变量 | 描述 | 类型 | 监控 | 权限 |
| Enabled | 插头是否可用 | Boolean | True | RW |
| Problem | 故障 | Boolean | True | R |
| Tripped | 触发故障原因 | OptionList | False | R |
| ACCurrent | 测量的AC电流 | Integer | False | R |
| PowerLimit | 最大功率限制 | Integer | False | RW |

## 5.3 Contactor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常见变量 | 描述 | 类型 | 监控 | 权限 |
| Active | 是否闭合 | Boolean | True | R |
| Problem | 开/关（失败） | Boolean | True | R |
| Tripped | 触发故障原因 | OptionList | False | R |

## 5.4 RadioLink

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常见变量 | 描述 | 类型 | 监控 | 权限 |
| Enabled | 是否启用网络 | Boolean | False | RW |
| Active | 无线网是否连上 | Boolean | False | R |
| IMSI | IMSI | String | False | R |
| ICCID | ICCID | String | False | R |
| IpAddress | 本机IP | String | False | R |
| ServerIp | 服务器IP | String | False | RW |
| ServerPort | 服务器端口号 | Integer | False | RW |
| Problem | 故障 | Boolean | False | R |
| WiredSignalStrength | 信号强度 | Integer | False | R |

## 5.5 TiltSensor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常见变量 | 描述 | 类型 | 监控 | 权限 |
| Enabled | 启用倾倒 | Boolean | True | RW |
| Active | 是否倾斜 | Boolean | True | R |

## 4.6 TemperatureSensor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常见变量 | 描述 | 类型 | 监控 | 权限 |
| Enabled | 启用温度传感器 | Boolean | True | RW |
| Problem | 故障 | Boolean | True | R |
| Temperature | 当前温度 | Integer | False | R |

## 4.7 Clock

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常见变量 | 描述 | 类型 | 监控 | 权限 |
| Active | 当前状态 | Boolean | False | R |
| SystemTicks | 上电到目前ticks | Integer | False | R |
| Time（Now） | 当前时间（CSMS seted） | DateTime | False | RW |
| TimeOffset | TimeZone | Integer | False | RW |
| NtpServerUri | Ntp服务器地址 | String | False | RW |

## 4.8 EVSE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常见变量 | 描述 | 类型 | 监控 | 权限 |
| Enabled | 启用/禁用EVSE | Boolean | True | RW |
| Problem | 故障 | Boolean | True | R |
| Tripped | 触发故障集合 | OptionList | True | R |
| ACVoltage | 测量的AC电压 | Integer | False | R |
| ACCurrent | 测量的AC电流 | Integer | False | R |
| DCVoltage | 测量的DC电压 | Integer | False | R |
| DCCurrent | 测量的DC电流 | Integer | False | R |
| ChargeProtocol | 充电协议 | OptionList | False | RW |