

# ferroamp

## Ferroamp External API

Revision E

2021-03-22



+46 (0)8 684 333 90



[www.ferroamp.com](http://www.ferroamp.com)



[support@ferroamp.se](mailto:support@ferroamp.se)



Domnarvsgatan 16. 16353 Spånga Sweden

1	Revision History	2
2	Disclaimer	4
3	Connection	4
3.1	Encryption	4
4	Data	5
4.1	Payload specification	5
4.1.1	Definitions	5
4.1.2	extapi/data/ehub	5
4.1.2.1	Example	7
4.1.3	extapi/data/eso	9
4.1.3.1	ESO Fault Codes	9
4.1.3.2	Example	10
4.1.4	extapi/data/sso	11
4.1.4.1	Example	11
4.1.5	extapi/data/esm	12
4.1.5.1	Example	12
5	Control	13
5.1	Request	13
5.1.1	Examples of request payloads	13
5.1.1.1.1	Charging with 5000W:	13
5.1.1.1.2	Discharging with 12 kW:	13
5.1.1.1.3	Returning control of batteries to system:	13
5.1.2	Power reference/argument in charge and discharge requests	13
5.2	Response	14
5.3	Result	14
6	Events	14

# 1 Revision History

Revision	History
E	<p>This document version applies to extapi version 1.2.1 and above or until a newer version is released</p> <p>New Features:</p> <ul style="list-style-type: none"> <li>Added explanations on some ESO fault codes</li> </ul>
D	<p>This document version applies to extapi version 1.2.0 and above or until a newer version is released</p> <p>New Features:</p> <ul style="list-style-type: none"> <li>Removed "parentEsoId" from ESM data</li> <li>Added "ratedPower" to ESM data</li> <li>Added explanation of "status" to ESM data</li> <li>Added "udc" to SSO data</li> <li>Added "udc" to ESO data</li> </ul> <p>Fixed issues:</p> <ul style="list-style-type: none"> <li>Missing description/definition of iace in ehub data</li> </ul>
C1	<p>Fixed Issues:</p> <ul style="list-style-type: none"> <li>Removed obsolete section "Confidential"</li> </ul>
C	<p>Fixed Issues:</p> <ul style="list-style-type: none"> <li>#684: Documentation: ExtApi Specification rev B has explanation of ESO Fault Codes in SSO section.</li> </ul>
B	<p>New Features:</p> <ul style="list-style-type: none"> <li>Added time stamps to data payloads</li> <li>Improved handling if charge/discharge requests as well as an updated explanation.</li> </ul> <p>Fixed issues:</p> <ul style="list-style-type: none"> <li>#652: Documentation: Update ExtApspec with explanation of fault codes for system, sso, eso</li> <li>#651: Documentation: ExtApi spec has incorrect unit on sext</li> <li>#645: ExtApi does not handle ESOs in master/slave configuration correctly</li> <li>#409: ESM data has flipped soc and soh (from extapi version 1.1.1)</li> <li>#388 : Extapi does not report system level SOC/SOH/Rated Capacity</li> </ul>
PB1	Available currents. Separate branch not described here
A1	Updated section 4.1

---

A

First formal release of specification

---

## 2 Disclaimer

The Ferroamp External API implementation is in beta release state and compatibility between different versions of this document is not guaranteed.

## 3 Connection

Data and commands are sent using mqtt where the user connects to the mqtt broker residing on the Ferroamp system.

Ip: address to unit  
Port: 1883  
Encrypted port: 8883

### 3.1 Encryption

Data shall be encrypted but for the moment we will use user/password.

User: to be sent separately  
Password: to be sent separately

## 4 Data

Data is published from a ferroamp system and may contain measurement data and system status. Data is sent periodically using mqtt.

Topics:

- extapi/data/ehub
- extapi/data/eso
- extapi/data/sso
- extapi/data/esm

Payload: json formatted data.

### 4.1 Payload specification

The payload is in json format specified as follows:

```
{ "parameter_name_1": { "value_representation_1" }, { "parameter_name_2":
{ "value_representation_2" }, ... "parameter_name_n":
{ "value_representation_n" }
```

The parameters can be required or optional. Optional may not be sent at all or sent intermittently

#### 4.1.1 Definitions

In the tables below the following abbreviations are used

Abbreviation	Meaning
<string>	ascii formatted string
<str_fl>	String representation of a float, i.e. "3.141528"
<str_uint8>	String representation of an unsigned 8-bit integer, i.e. "1"
<str_uint16>	String representation of an unsigned 16-bit integer, i.e. "514"
<str_uint32>	String representation of an unsigned 32-bit integer, i.e. "410123"
<str_uint64>	String representation of an unsigned 64-bit integer "2454683685"

#### 4.1.2 extapi/data/ehub

Transmission interval: 1s

Parameter name	Value representation	required	Unit of measurement	Comment
'gridfreq'	{ "val": <str_fl> }	no	Hz	Estimated Grid Frequency
'ul'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	V	External voltage
'iace'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	ACE equalization current set-points in Arms
'il'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	Inverter RMS current
'ild'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	Inverter reactive current
'ilq'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	Inverter active current
'iext'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	External/grid RMS current
'iextd'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	External/grid reactive current
'iextq'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	External/grid active current
'iLoadd'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	
'iLoadq'	{ "L1": <str_fl>, "L2": <str_fl>, "L3": <str_fl> }	yes	A	

'soc'	{"val":<str_fl>}	yes	%	State Of Charge for the system
'soh'	{"val":<str_fl>}	yes	%	State Of Health for the system
'sext'	{"val":<str_fl>}	yes	VA	apparent power
'pext'	{"L1":<str_fl>,"L2":<str_fl>,"L3":<str_fl>}	yes	W	external/grid power, active
'pextreactive'	{"L1":<str_fl>,"L2":<str_fl>,"L3":<str_fl>}	yes	W	external/grid power, reactive
'pinv'	{"L1":<str_fl>,"L2":<str_fl>,"L3":<str_fl>}	yes	W	inverter power, active
'pinvreactive'	{"L1":<str_fl>,"L2":<str_fl>,"L3":<str_fl>}	yes	W	inverter power, active
'pload'	{"L1":<str_fl>,"L2":<str_fl>,"L3":<str_fl>}	yes	W	
'ploadreactive'	{"L1":<str_fl>,"L2":<str_fl>,"L3":<str_fl>}	yes	W	
'ppv'	{"val":<str_fl>}	no	W	Only sent when system has PV
'pbat'	{"val":<str_fl>}	no	W	Only sent when system has batteries
'ratedcap'	{"val":<str_fl>}	no	Wh	Total rated capacity of all batteries in system
'wextprodq'	{"L1":<str_uint64>,"L2":<str_uint64>,"L3":<str_uint64>}	yes	mJ	
'wextconsq'	{"L1":<str_uint64>,"L2":<str_uint64>,"L3":<str_uint64>}	yes	mJ	
'winvprodq'	{"L1":<str_uint64>,"L2":<str_uint64>,"L3":<str_uint64>}	yes	mJ	
'winvconsq'	{"L1":<str_uint64>,"L2":<str_uint64>,"L3":<str_uint64>}	yes	mJ	
'wloadprodq'	{"L1":<str_uint64>,"L2":<str_uint64>,"L3":<str_uint64>}	yes	mJ	
'wloadconsq'	{"L1":<str_uint64>,"L2":<str_uint64>,"L3":<str_uint64>}	yes	mJ	
'wpv'	{"val":<str_uint64>}	no	mJ	Only sent when system has PV
'wbatprod'	{"val":<str_uint64>}	no	mJ	Only sent when system has batteries
'wbatcons'	{"val":<str_uint64>}	no	mJ	Only sent when system has batteries
'state'	{"val":<str_uint32>}	yes	bit values	State of the system
'udc'	{"neg":<str_fl>,"pos":<str_fl>}	yes	V	Positive and negative DC Link voltage
'ts'	{"val":<string>}	yes	Format: "YYYY-MM-DDThh:mm:ssUTC"	Time stamp when message was published.

## 4.1.2.1 Example

```

{
  "gridfreq": {
    "val": "49.95"
  },
  "iace": {
    "L1": "0.00", "L2": "0.00", "L3": "0.00"
  },
  "iext": {
    "L1": "3.66", "L2": "3.05", "L3": "3.53"
  },
  "iextd": {
    "L1": "1.01", "L2": "1.84", "L3": "1.93"
  },
  "iextq": {
    "L1": "4.06", "L2": "3.84", "L3": "4.18"
  },
  "il": {
    "L1": "2.00", "L2": "1.97", "L3": "1.97"
  },
  "ild": {
    "L1": "2.64", "L2": "2.60", "L3": "2.59"
  },
  "ilq": {
    "L1": "4.06", "L2": "3.84", "L3": "4.18"
  },
  "iload": {
    "L1": "-1.63", "L2": "-0.76", "L3": "-0.66"
  },
  "iloadd": {
    "L1": "-1.63", "L2": "-0.76", "L3": "-0.66"
  },
  "iloadq": {
    "L1": "3.94", "L2": "3.75", "L3": "4.07"
  },
  "ilq": {
    "L1": "0.12", "L2": "0.09", "L3": "0.11"
  },
  "pbat": {
    "val": "0.00"
  },
  "pext": {
    "L1": "686.33", "L2": "643.69", "L3": "705.32"
  },
  "pextreactive": {
    "L1": "169.47", "L2": "308.67", "L3": "326.25"
  },
  "pinv": {
    "L1": "20.21", "L2": "15.18", "L3": "18.61"
  },
  "pinvreactive": {
    "L1": "444.69", "L2": "434.94", "L3": "436.47"
  },
  "pload": {
    "L1": "666.13", "L2": "628.51", "L3": "686.71"
  },
  "ploadreactive": {
    "L1": "-275.22", "L2": "-126.28", "L3": "-110.22"
  },
  "ppv": {
    "val": "0.00"
  },
  "ratedcap": {
    "val": "7200.00"
  },
  "sext": {
    "val": "2435.31"},
  "soc": {

```



```
    "val": "41.04"
  },
  "soh": {
    "val": "95.82"
  },
  "state": {
    "val": "4096"
  },
  "ts": {
    "val": "2019-01-18T14:23:10UTC"
  },
  "udc": {
    "neg": "-379.91", "pos": "380.30"
  },
  "ul": {
    "L1": "239.07", "L2": "237.24", "L3": "239.06"
  },
  "wbatcons": {
    "val": "415000012234"
  },
  "wbatprod": {
    "val": "588379204908"
  },
  "wextconsq": {
    "L1": "5654992911585", "L2": "3161182619121", "L3": "5504478918590"
  },
  "wextprodq": {
    "L1": "939740958160", "L2": "1513371098182", "L3": "1040432778248"
  },
  "winvconsq": {
    "L1": "857222905812", "L2": "742769639789", "L3": "609409141152"
  },
  "winvprodq": {
    "L1": "1867481874316", "L2": "1743008187583", "L3": "2179580125420"
  },
  "wloadconsq": {
    "L1": "8097779271506", "L2": "4156753076161", "L3": "8472279880275"
  },
  "wloadprodq": {
    "L1": "544456452195", "L2": "193311772496", "L3": "429837473775"
  },
  "wpv": {
    "val": "9673466178757"
  }
}
```

### 4.1.3 extapi/data/eso

Transmission interval: Default 5s. Configurable between 1 and 30 seconds.

Parameter name	Value representation	required	Unit of measurement	Comment
'id'	{"val":<string>}	yes		Unique identifier
'ubat'	{"val":<str_fl>}	yes	V	Measured on battery side
'ibat'	{"val":<str_fl>}	yes	A	Measured on battery side
'wbatprod'	{"val":<str_uint64>}	yes	mJ	Total energy produced by ESO, i.e total energy <b>discharged</b>
'wbatcons'	{"val":<str_uint64>}	yes	mJ	Total energy consumed by ESO, i.e total energy <b>charged</b>
'soc'	{"val":<str_fl>}	yes	%	State of Charge for ESO
'relaystatus'	{"val":<str_uint8>}	yes		0 = relay closed 1 = relay open
'temp'	{"val":<str_fl>}	yes	degrees Celsius	Measured inside ESO
'faultcode'	{"val":<str_uint16>}	yes	bitmask	See section 4.1.3.1 below for further explanations on Fault Codes.
'udc'	{"val":<str_fl>}	yes	V	DC link voltage as measured by ESO
'ts'	{"val":<string>}	yes	Format: "YYYY-MM-DDThh:mm:ssUTC"	Time stamp when message was published.

#### 4.1.3.1 ESO Fault Codes

The ESO can report a number of fault codes, Fault code is on bit format and each flag/bit is described below:

Bit (uint16, MSB)	Hex value	Comment
0	0x01	The pre-charge from battery to ESO is not reaching the voltage goal prohibiting the closing of the relays.
1	0x02	CAN communication issues between ESO and battery.
2	0x04	This indicates that the SoC limits for the batteries are not configured correctly, please contact Ferroamp Support for help.
3	0x08	This indicates that the power limits for the batteries are incorrect or non-optimal.  When controlling batteries via extapi and the system is set in either peak-shaving or self-consumption modes this flag may be set but it will <b>not</b> affect control.  When <b>not</b> controlling batteries via extapi this indicates that the settings made in EMS Configuration is invalid.
4	0x10	On-site emergency stop has been triggered.
5	0x20	The DC-link voltage in ESO is so high that it prevents operation.
6	0x40	Indicates that the battery has an alarm or an error flag raised. Please check Battery manufacturer's manual for trouble shooting the battery, or call Ferroamp Support.
7	0x80	Not a fault, just an indication that Battery Manufacturer is not Ferroamp
8	Not used	
9	Not used	
10	Not used	
11	Not used	
12	Not used	
13	Not used	
14	Not used	
15	Not used	

Bit 7 can safely be ignored, for all other flags please contact Ferroamp support.

When running in either "Self-Consumption" or "Peak Shaving" modes flag 3 can also be ignored if batteries are controlled via extapi.

For all Fault Codes above, please contact Ferroamp Support for help in troubleshooting.

#### 4.1.3.2 Example

```
{
  "faultcode": {
    "val": "0"
  },
  "id": {
    "val": "17080008"
  },
  "ibat": {
    "val": "1.00"
  },
  "ubat": {
    "val": "470.0"
  },
  "relaystatus": {
    "val": "1"
  },
  "soc": {
    "val": "100.0"
  },
  "temp": {
    "val": "24.47"
  },
  "wbatcons": {
    "val": "1027307459944"
  },
  "wbatprod": {
    "val": "1027307459944"
  },
  "udc": {
    "val": "760.0"
  },
  "ts": {
    "val": "2019-01-18T14:23:10UTC"
  }
}
```

#### 4.1.4 extapi/data/sso

Transmission interval: Default 5s. Configurable between 1 and 30 seconds.

Parameter name	Value representation	required	Unit of measurement	Comment
'id'	{"val":<string>}	yes		Unique identifier of SSO
'upv'	{"val":<str_fl>}	yes	V	measured on PV string side
'ipv'	{"val":<str_fl>}	yes	A	measured on PV string side
'wpv'	{"val":<str_uint64>}	yes	mJ	Total energy produced by SSO
'faultcode'	{"val":<str_uint16>}	yes	bitmask	0x00 = OK For all other values please contact Ferroamp support
'relaystatus'	{"val":<str_uint8>}	yes		0 = relay closed (i.e running power) 1 = relay open/disconnected 2 = precharge
'temp'	{"val":<str_fl>}	yes	degrees Celsius	Temperature measured on PCB of SSO
'udc'	{"val":<str_fl>}	yes	V	DC link voltage as measured by SSO
'ts'	{"val":<string>}	yes	Format: "YYYY-MM-DDThh:mm:ssUTC"	Time stamp when message was published.

##### 4.1.4.1 Example

```
{
  "relaystatus": {
    "val": "1"
  },
  "temp": {
    "val": "24.47"
  },
  "wpv": {
    "val": "1027307459944"
  },
  "faultcode": {
    "val": "0"
  },
  "ipv": {
    "val": "0"
  },
  "upv": {
    "val": "0.43"
  },
  "id": {
    "val": "17080008"
  },
  "udc": {
    "val": "760.0"
  },
  "ts": {
    "val": "2019-01-18T14:23:10UTC"
  }
}
```

### 4.1.5 extapi/data/esm

Transmission interval: Default 60 s. Configurable between 1 and 300 seconds.

<i>Parameter name</i>	<i>Value representation</i>	<i>required</i>	<i>Unit of measurement</i>	<i>Comment</i>
'id'	{"val":<string>}	yes		Unique identifier of battery. If available, this will be the unique id that the battery reports.
'soh'	{"val":<str_fl>}	yes	%	
'soc'	{"val":<str_fl>}	yes	%	
'ratedCapacity'	{"val":<str_fl>}	yes	Wh	Rated capacity of battery
'ratedPower'	{"val":<str_fl>}	yes	W	Rated power of battery
'status'	{"val":<str_uint16>}	yes	bitmask	Dependent on battery manufacturer
'ts'	{"val":<string>}	yes	Format: "YYYY-MM-DDThh:mm:ssUTC"	Time stamp when message was published.

#### 4.1.5.1 Example

```
{
  "soh": {
    "val": "100.0"
  },
  "soc": {
    "val": "100.0"
  },
  "ratedCapacity": {
    "val": "7200.0"
  },
  "id": {
    "val": "17020004"
  },
  "ratedPower": {
    "val": "7000.0"
  },
  "status": {
    "val": "0"
  },
  "ts": {
    "val": "2019-01-18T14:23:10UTC"
  }
}
```

## 5 Control

To control the ferroamp system publish a message on the topic /extapi/request. The system will validate the settings, execute them if possible and return the changed settings on topic /extapi/result. The system only allows for one request at a time until the whole sequence of request-response-result is done no new requests will be acted upon. If a new request is sent before the current is finished the ehub will return a NAK and a message stating that a request/transaction is already in progress.

Topics:

- extapi/control/request
- extapi/control/response
- extapi/control/result

### 5.1 Request

Request payload is on json format and has two predefined keys:

```
{"transId":<str>,"cmd":{cmdValue}}
```

Parameter name	Value representation	required	Comment
'transId'	<string>	yes	Unique identifier of transaction (i.e. incremented integer, UUID)
'cmd'	{<name>:<string>,<arg_1>:<value_1>,... {<arg_n>:<arg_n>}}	yes	Command data in json format. Contains one or more key-value pairs. See specification below

Below is a list of supported requests and their arguments

Command name	Value representation	Argument1 name	Argument 1 repr	Comment
'charge'	<string>	"arg"	<string>	arg is charge power in W
'discharge'	<string>	"arg"	<string>	arg is discharge power in W
'auto'	<string>	N/A	N/A	No argument needed
'extapiversion'	<string>	N/A	N/A	No argument needed. Version no is returned in response, no result sent

Future commands to be added as needed, e.g. "soclimits"

#### 5.1.1 Examples of request payloads

5.1.1.1.1 Charging with 5000W:

```
extapi/control/request {"transId":"1",
cmd":{"name":"charge","arg":"5000"}}
```

5.1.1.1.2 Discharging with 12 kW:

```
extapi/control/request {"transId": "1508459760", "cmd":{"
name":"discharge","arg":" 12000"}}
```

5.1.1.1.3 Returning control of batteries to system:

```
extapi/control/request {"transId": "989C6E5C-2CC1-11CA-A044-
08002B1BB4F5", "cmd": {"name": "auto"}}
```

#### 5.1.2 Power reference/argument in charge and discharge requests

The argument is power reference for the system and not the power reference of each respective battery.

When received the system will evaluate the current rated capacity and maximum charge/discharge power limits of the system. Depending on the status reported by ESOs and batteries the current rated capacity and power limits may differ from the system's nominal capacity and power limits. For instance, if one battery is not available due to a fault the current rated capacity and power limits of the system will be reduced by the nominal capacity and power of that battery.

## 5.2 Response

Response carries data on the status of the request and, if possible, and explanation on the status. The response and status are sent directly as a response to request and shall be seen as an indication of whether the request is applicable or not. If applicable the result of the actual request will be sent in a "result" topic, see 5.3 below. transId value will be the same as in the corresponding request payload.

Examples:

```
{"status": "ack", "msg": "sending cmd to ESOs", "transId": "1"}
```

```
{"status": "nak", "msg": "Other transaction in progress", "transId": "128"}
```

```
{"status": "nak", "msg": "Max allowed power is 24000 W", "transId": "311"}
```

## 5.3 Result

Carries the information on the result of the request. It is sent asynchronously when the request has been executed. transId value will be the same as in the corresponding request payload.

```
{"transId":<string>, "status":<ack/nak>, "msg": <string>}
```

## 6 Events

To be implemented

The ferroamp system has the possibility to publish event data on asynchronous events.

Topics:

- extapi/control/event/ {"timestamp":utc date, "event": {<string>}}