Description of the structures and variables of the Smart Contract

Objective

This document presents and describes the variables, structures and methods present in the smart contract.

Structures

- METERING

Structure responsible for grouping the information of readings.

<table>
<tr><th>Nome da Variável</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td>Typemeter</td><td>uint</td><td>Indicates the type of reading

<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Total Energy</td></tr><tr><td>2</td><td>Peak Energy</td></tr><tr><td>3</td><td>Off – Peak Energy</td></tr><tr><td>4</td><td>Reserved Energy</td></tr><tr><td>30</td><td>Current</td></tr><tr><td>31</td><td>Location</td></tr><tr><td>32</td><td>Battery</td></tr></table></td></tr>
<tr><td>Valuedata</td><td>String</td><td>Metering Data Value</td></tr>
<tr><td>Timestamp</td><td>uint</td><td>timestamp according to UNIX</td></tr>
</table>

Exemples:

1) Sending Total Energy

Metering	
TyperMeter	1
ValueData	“15358”
TimeStamp	1612199928

2) Sending Peak Energy

Metering	
TyperMeter	2
ValueData	“4500”
TimeStamp	1612199975

3) Sending Off-peak Energy

Metering	
TyperMeter	3
ValueData	“1373”
TimeStamp	1612199940

4) Sending Location

Metering	
TyperMeter	30
ValueData	“23.622460231714374, -46.629647445891116”
TimeStamp	1612191057

5) Sending Location

Metering	
TyperMeter	32
ValueData	“67”
TimeStamp	1612191057

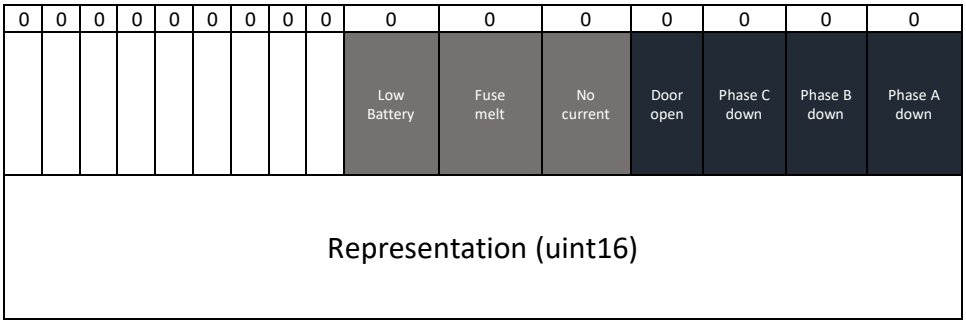
EVENT METER:

Structure responsible for grouping the information of an event

Nome da variável	Tipo	Descrição
Typeevent	uint	Indicates the type of event that occurred
Timestamp	uint	timestamp according to UNIX

Information about TypeEvent

Each event is represented by a bit within a uint32 variable, when an event occurs, the respective bit takes on a value of 1. The table below shows how the events are structured within the variable



For example, if the phase C is down and the door open events occur, we have the following binary value: $1100_{(2)}$ being represented as 12 in decimal or C in hexadecimal

As another example, suppose the sensor fuse is blown, the following event value will be generated (binary, decimal, hexadecimal) respectively: $10000_{(2)}$, $32_{(10)}$, 0x20

Exemples:

1) Sending phase C down and door open event

EventMeter	
TypeEvent	12
TimeStamp	1612199928

2) Sending phase A down

EventMeter	
TypeEvent	1
TimeStamp	1612199928

3) Sending door open event

EventMeter	
TypeEvent	8
TimeStamp	1612199928

METER:

Structure responsible for representing a meter.

Nome da variável	Tipo	Descrição
Serialnumber	String	Indicates the meter serial number
Activated	Boolean	Indicates the status of the meter
Interval	uint	Indicates the reading range
Timestamp	uint	Indicates the timestamp according to the UNIX standard
Events	Eventmeter[]	Indicates registered events
EventsCount	uint	Indicates the number of events present in the structure
Meterings	Metering[]	Indicates the readings performed
MeteringsCount	uint	Indicates the amount of readings present in the structure

Methods present in the contract:

- ***setMeterup*** *(address meter, string memory serialnumber, unit timestamp):*

  Description: Saves the information that a meter has been started

- ***setMeterdown*** *(address meter, uint timestamp):*

  Description: Saves the information that a meter has been turned off

- ***setMeterInterval*** *(address meter, uint interval, uint timestamp):*

  Description: Sets the meter reading range;

- ***setMeterEvent****(address meter, uint typeevent, uint timestamp):*

  Description: Saves the event occurred with a measuring device;

- **setMeterData**(address meter, uint typemeter, uint valuedata, uint timestamp):

  Description: Registers a reading that occurred on the meter device.

- **getMeterData**(address device):

  Description: Retrieve meter information

- **getMeterEventData**(address device, uint index):

  Description: Retrieves the event with the desired index.

- **getMeterMeteringData**(address device, uint index):

  Description: Retrieves the Metering structure of the desired device and index