**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.

|  |  |  |
| --- | --- | --- |
| Nome da Variável | Tipo | Descrição |
| Typemeter | uint | Indicates the type of reading   |  |  | | --- | --- | | **Value** | **Description** | | 1 | Total Energy | | 2 | Peak Energy | | 3 | Off – Peak Energy | | 4 | Reserved Energy | | 30 | Current | | 31 | Location | | 32 | Battery | |
| Valuedata | String | Metering Data Value |
| Timestamp | uint | timestamp according to UNIX |

**Exemples:**

1. Sending Total Energy

|  |  |
| --- | --- |
| **Metering** | |
| **TyperMeter** | 1 |
| **ValueData** | “15358” |
| **TimeStamp** | 1612199928 |

1. Sending Peak Energy

|  |  |
| --- | --- |
| **Metering** | |
| **TyperMeter** | 2 |
| **ValueData** | “4500” |
| **TimeStamp** | 1612199975 |

1. Sending Off-peak Energy

|  |  |
| --- | --- |
| **Metering** | |
| **TyperMeter** | 3 |
| **ValueData** | “1373” |
| **TimeStamp** | 1612199940 |

1. Sending Location

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

1. Sending Location

|  |  |
| --- | --- |
| **Metering** | |
| **TyperMeter** | 32 |
| **ValueData** | “67” |
| **TimeStamp** | 1612191057 |

**EVENT METER:**

Estrutura responsável por agrupar as informações de um evento

|  |  |  |
| --- | --- | --- |
| 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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |  | Low Battery | Fuse  melt | No current | Door open | Phase C down | Phase B down | Phase A down |
| Representation (uint16) | | | | | | | | | | | | | | | |

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 |

1. Sending phase A down

|  |  |
| --- | --- |
| **EventMeter** | |
| **TypeEvent** | 1 |
| **TimeStamp** | 1612199928 |

1. 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):*

**Descrição:**Saves the information that a meter has been started

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

**Descrição:**Saves the information that a meter has been turned off

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

**Descrição**: Sets the meter reading range;

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

**Descrição**: Saves the event occurred with a measuring device;

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

**Descrição**: Registers a reading that occurred on the meter device.

* **getMeterData**(address device):

**Descrição**: Retrieve meter information

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

**Descrição**: Retrieves the event with the desired index.

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

**Descrição**: Retrieves the Metering structure of the desired device and index