
Prism

Release 0.0.1

Sep 27, 2019

Contents

1	1- Drilling template	1
2	Pinout	3
2.1	Top view	3
2.2	Bottom view	3
2.3	Front view	6
2.4	Connectors	6
3	Connection Diagram	13
4	Protocol definition for EVSEsd over ubus	15
4.1	Object “evse.control”	15
5	Color codes	21
5.1	Button green	22
5.2	Button red	24
5.3	Button yellow	25
5.4	Button purple	25
5.5	Button light blue	26
5.6	Others	26

CHAPTER 1

1- Drilling template

Download the drilling template [here](#).

CHAPTER 2

Pinout

Here you can find all the connectors pinout.

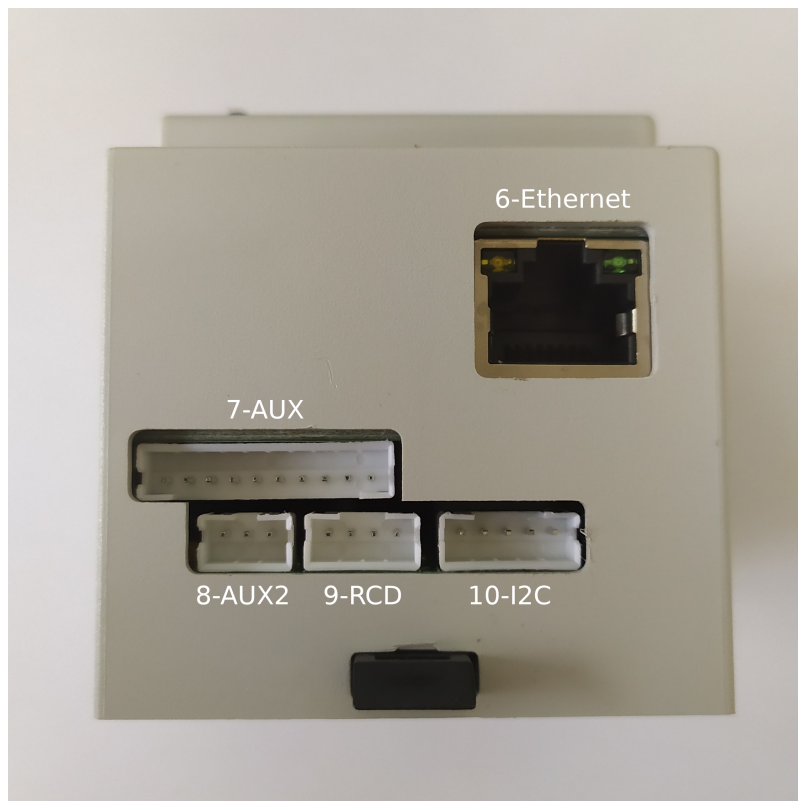
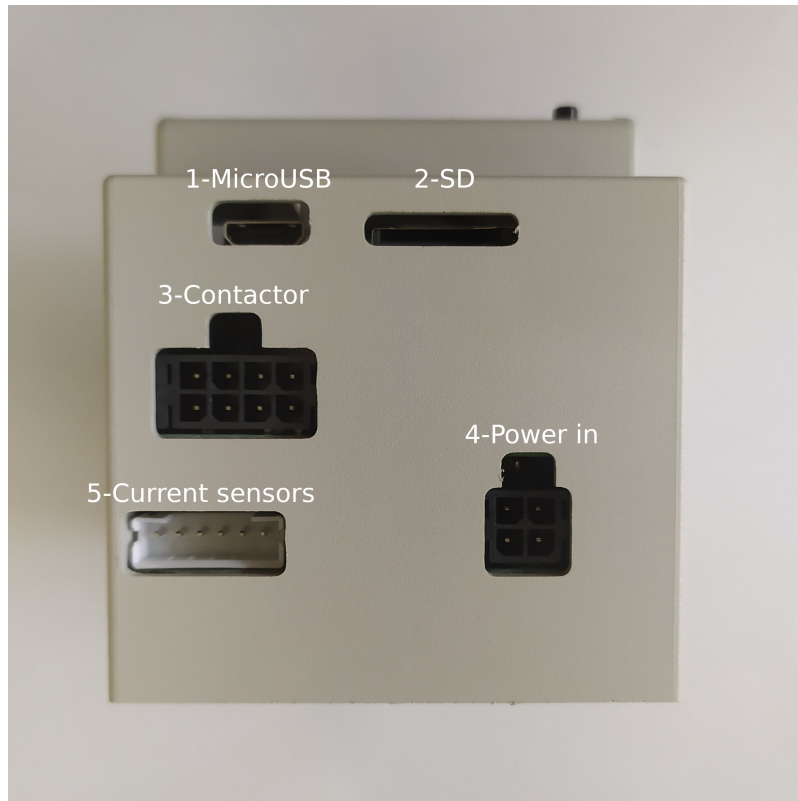
Note: All the part numbers are for the mating connectors

2.1 Top view

Number	Description	Part Number
1	<i>Micro USB host</i>	
2	<i>SD</i>	
3	<i>Contactator</i>	Wurt Elektronik - 662008113322
4	<i>Power input</i>	Wurt Elektronik - 662004113322
5	<i>Current sensors</i>	JST - PHR-6

2.2 Bottom view

Number	Description	Part Number
6	<i>Ethernet</i>	
7	<i>AUX</i>	JST - PHR-10
8	<i>AUX2</i>	JST - PHR-3
9	<i>RCD</i>	JST - PHR-4
10	<i>I2C</i>	JST - PHR-5





2.3 Front view

Number	Description	Part Number
11	<i>Front Panel</i>	JST - PHR-5

2.4 Connectors

2.4.1 1. Micro USB host

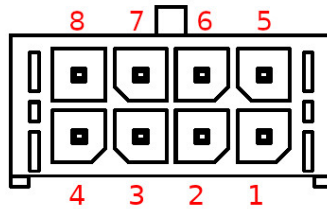
Warning: Max 250 mA

2.4.2 2. SD

Micro SD slot to expand storage

2.4.3 3. Contactor

Drive contactors and check stuck condition.

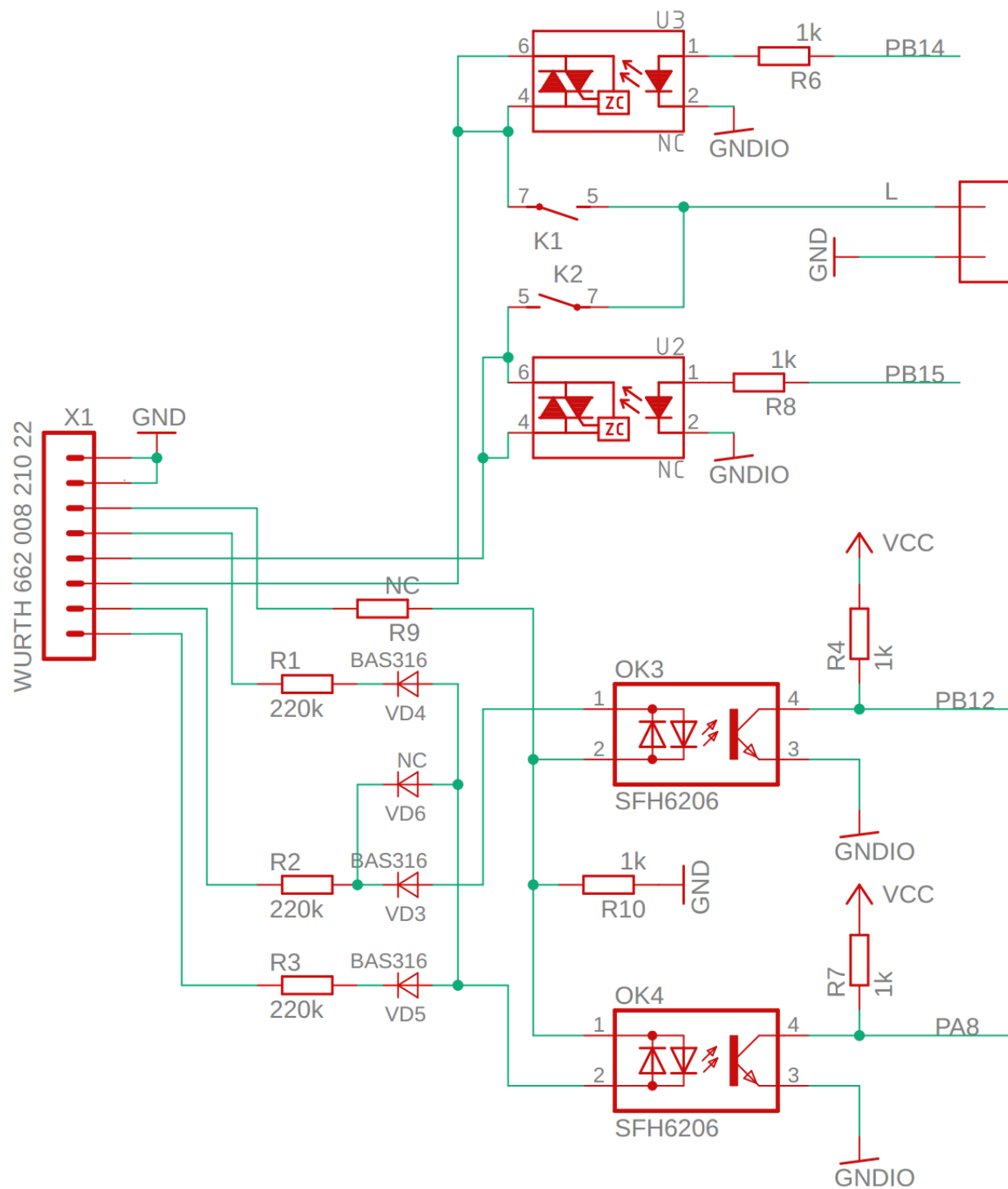


Pin	Signal	Description
1	N	Neutral
2	N	Neutral
3	Ns	Neutral out sense (not used)
4	LAs1	Line out sense A input 1
5	CL2	Contactactor 2 Live terminal
6	CL1	Contactactor 1 Live terminal
7	LBs1	Line out sense B input 1
8	LAs2	Line out sense A input 2

LAs1, LAs2 and LBs1 are connected to optocouplers to detect high voltages at the output of the contactor, to detect a welded contact.

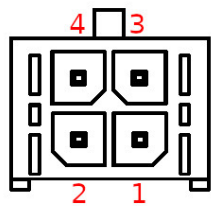
Contactactor coils can be driven by internal relays output. Connect the main one on pin 2 and 6, and the secondary one to pins 1 and 5.

Check the [Connection diagram](#) for more details



2.4.4 4. Power input

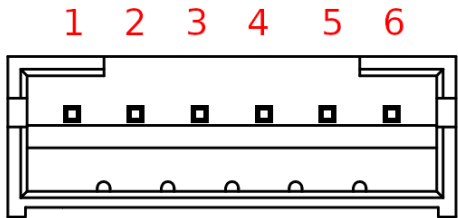
Connects to neutral and all phases available (1 or 3).



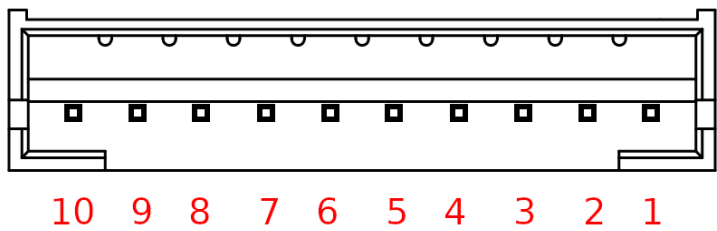
Pin	Signal	Description
1	L3	Phase 3
2	L2	Phase 2
3	N	Neutral
4	L1	Phase 1

2.4.5 5. Current sensors

Connects to the TA current sensor(s).



Pin	Signal	Description
1	A_P	Phase A TA input +
2	A_N	Phase A TA input -
3	B_P	Phase B TA input +
4	B_N	Phase B TA input -
5	C_P	Phase C TA input +
6	C_N	Phase C TA input -

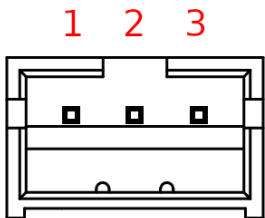


2.4.6 6. Ethernet

2.4.7 7. AUX

Pin	Signal	Description
1	PO2	Pilot signal out 2
2	PO1	Pilot signal out 1
3	RSB	RS485 -
4	RSA	RS485 +
5	CANH	CAN high
6	CANL	CAN low
7	LEDOUT	Programmable digital output, internally pulled up
8	GND	Ground
9	GND	Ground
10	VCC	5V

2.4.8 8. AUX2

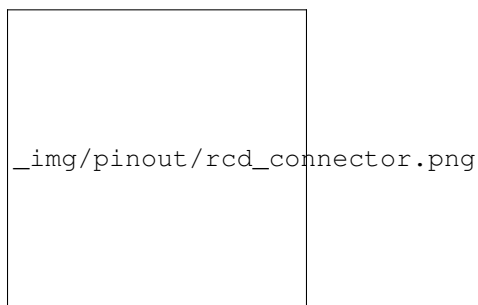
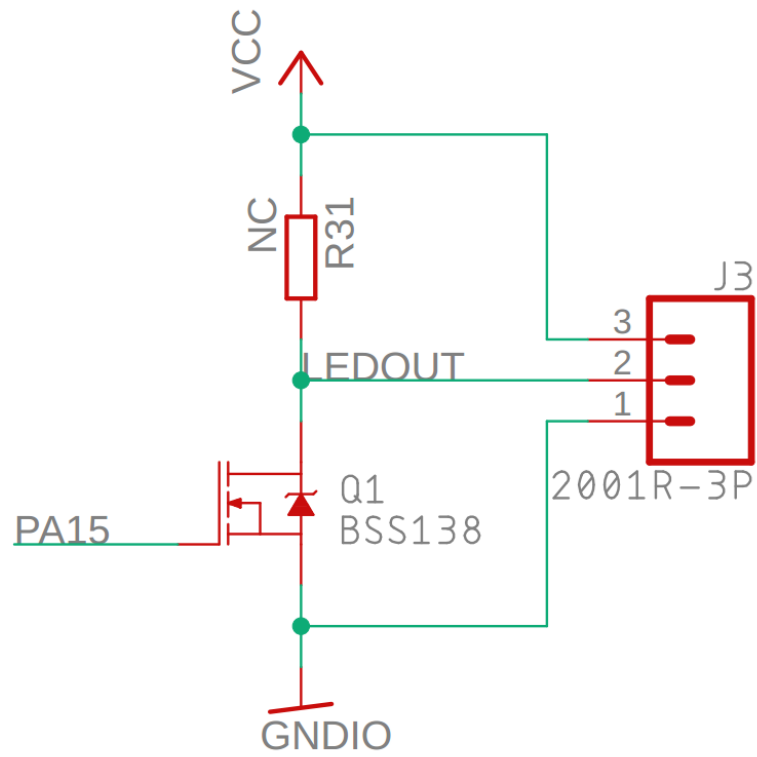


Pin	Signal	Description
1	VCC	5V
2	LEDOUT	Programmable digital output, internally pulled up
3	GND	Ground

Connector internal schematic:

2.4.9 9. RCD

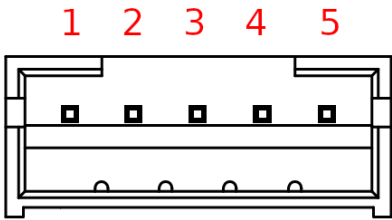
Connects to an RCM14-03 residual current monitor.



Pin	Signal	Description
1	GND	Ground
2	+12V	12V
3	TEST	RCD test output
4	RCD_FAULT	RCD fault input

2.4.10 10. I2C

I2C bus for communications. Internal 4K7 resistors.

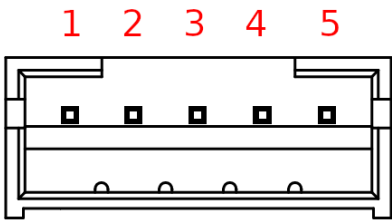


Pin	Signal	Description
1	NC	-
2	SCL	I2C SCL
3	SDA	I2C SDA
4	3V3	3.3V
5	GND	Ground

2.4.11 11. Front Panel

Connects to front panel.

Warning: All logic must be at 3.3V

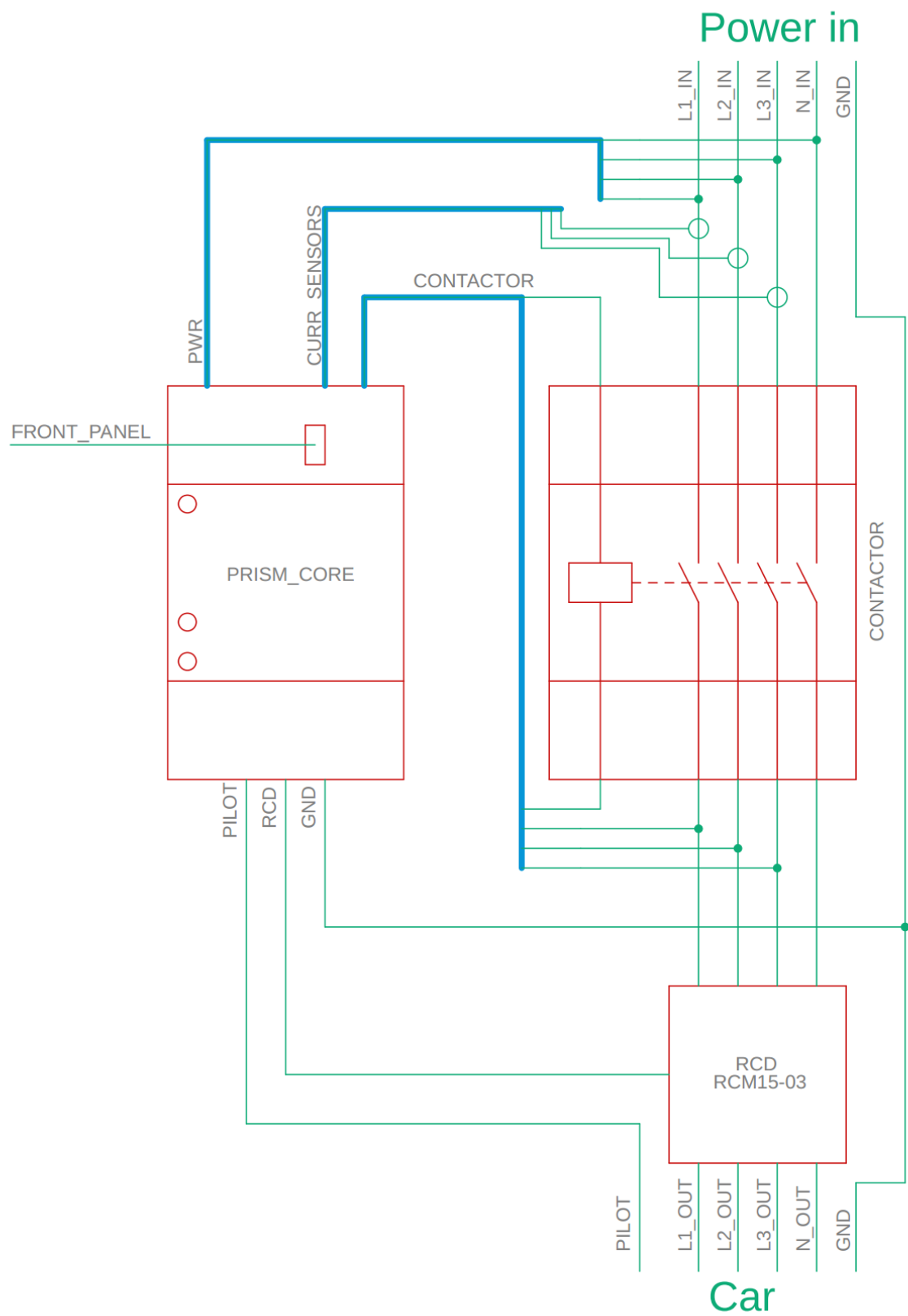


Pin	Signal	Description
1	RST	Reset
2	RX	Serial receive
3	TX	Serial transmit
4	5V	5V
5	GND	Ground

CHAPTER 3

Connection Diagram

Here is the basic connection diagram for the Prism Core



Protocol definition for EVSEsd over ubus

4.1 Object “evse.control”

Port states definition

State	Description
<i>idle</i>	...
<i>waiting</i>	...
<i>charging</i>	...

4.1.1 Command “reload”

Description:

Forces a reading of the current state for each charging port.

Request parameters:

None

Response:

None

4.1.2 Command “read_temperatures”

Description:

Forces a reading of the all the temperatures sensors for the charging station.

Request parameters:

None

Response:

None

4.1.3 Command “set_port_info”

Description:

Sets permanent port configuration parameters.

!!DANGER!! This command sets critical port parameters and must be used only by authorized installers.

Request parameters:

- `port` *integer* Port number starting from 1
- `current_max_limit` *float* Maximum configurable current in A
- `opt_autostart` *bool* Whether the port automatically starts charging when connected
- `opt_singlephase` *bool* Whether the port only supports single phase charging mode

Response:

None

Sample request:

```
{
    "port": 1,
    "current_max_limit": 15.90000,
    "opt_autostart": true,
    "opt_singlephase": false
}
```

4.1.4 Command “get_status”

Description:

Requests the global state of the charging station.

Request parameters:

None

Response:

- `firmware` *string* Main firmware version
- `stm-firmware` *string* Internal STM32 firmware version
- `stm-cpuid` *string* Internal STM32 CPU id
- `ports` *array* List of ports with their current state (see below)
- `temperatures` *array* List of temperatures reading for various internal sensors
- `error` *integer* Global error state code for the charging station

Single port record definition:

- `port` *integer* Port number starting from 1
- `status` *string* Port state
- `current_limit` *float* Maximum configurable current in A
- `current_max` *float* Maximum charging current in A
- `current_now` *float* Actual current in A
- `voltage_now` *float* Actual voltage in V
- `power_now` *integer* Actual power in W
- `energy_session` *integer* Total energy charged in this session in Wh
- `energy_total` *integer* Total energy charged lifetime in Wh
- `session_time` *integer* Elapsed time for the current charging session in seconds
- `connected` *bool* Indicates if a vehicle is connected to the port
- `threephase` *bool* Indicates if the port is currently using threephase
- `error` *bool* Indicates if the port is in error

Sample response:

```
{
  "status": "success",
  "result": {
    "firmware": "Prism 1.0 build 492",
    "stm-firmware": "0.1",
    "stm-cpuuid": "12FF4E123E1D001234567890",
    "ports": [
      {
        "port": 1,
        "status": "charging",
        "current_limit": 32.000000,
        "current_max": 6.000000,
        "current_now": 4.800000,
        "voltage_now": 232.500000,
        "power_now": 10.000000,
        "energy_session": 840.000000,
        "energy_total": 92400.000000,
        "session_time": 928,
        "connected": true,
        "threephase": false,
        "error": 0
      }
    ],
    "temperatures": [
      45,
      null,
      null,
      null
    ],
    "error": 0
  }
}
```

4.1.5 Command “set_current”

Description:

Sets the maximum current that can be charged on a specific port.

Request parameters:

- `port` *integer* Port number starting from 1
- `current_max` *float* Maximum charging current in A

Response:

None

Sample request:

```
{
    "port": 2,
    "current_max": 22.500000
}
```

4.1.6 Event “port_current_max_changed”

Description:

Notifies that the maximum charging current for a port has changed.

Event parameters:

- `port` *integer* Port number starting from 1
- `current_max` *float* Maximum charging current in A

Sample event:

```
{
    "port": 1,
    "current_max": 8.000000
}
```

4.1.7 Event “port_connected_change”

Description:

Notifies that a vehicle has been connected or disconnected from a charging port.

Event parameters:

- `port` *integer* Port number starting from 1
- `connected` *bool* Indicates if a vehicle is connected to the port

Sample event:

```
{  
    "port": 1,  
    "connected": true  
}
```

4.1.8 Event “error_change”

Description:

Notifies that an error situation has appeared or disappeared.

*Event parameters:**

- `port integer` Optional port number starting from 1 (zero means a global error)
- `error integer` Error code

Sample event:

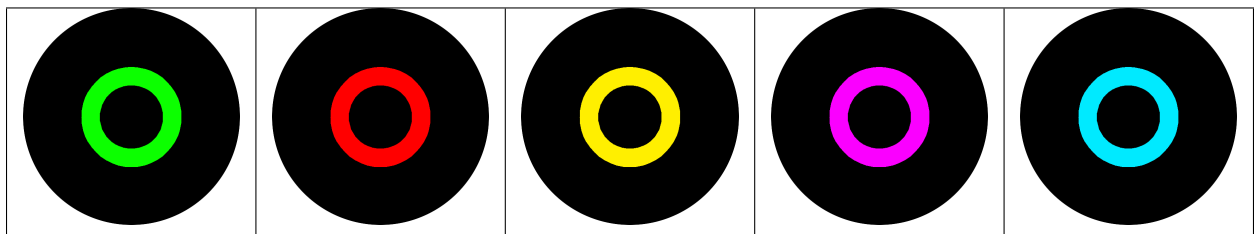
```
{  
    "port": 0,  
    "error": 80  
}
```


CHAPTER 5

Color codes

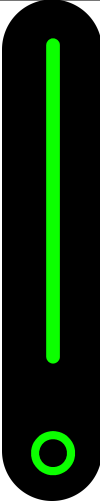
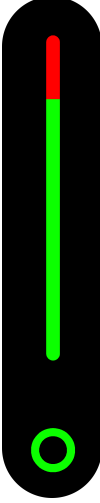
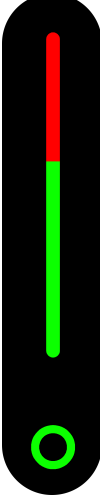
Prism leds are used to signal all different mode of function and error codes. Here is all the possible color codes available.

For simpler search thorough all codes select first the button's color:

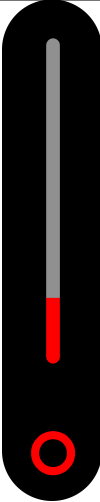
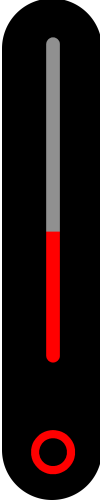
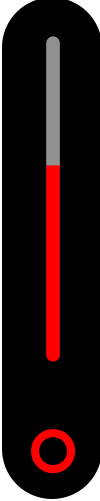
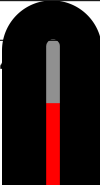


Note: “Can’t find your color? Try [this section](#) for other color codes”

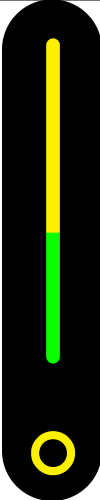
5.1 Button green

Color	Name	Description
	EV charging:	EV charging from the main power
	Overheating	Overheating, EV charge current is limited
	Overcurrent	Current above the set value (< +10%) but under maximum limit. EV charging continues.

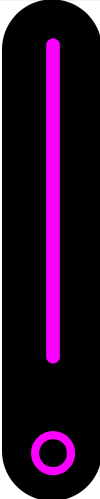
5.2 Button red

Color	Name	Description
	Overheat	Temperature above limit, waiting for cooling
	EV error	EV not recognized, charging not available
	Overcurrent	Current above the set value (> +10%) or above maximum limit. EV charging stops.
		Chapter 5. Color codes


5.3 Button yellow

Color	Name	Description
	Solar charging	<p>Charging with some power coming from PV panels. The yellow leds indicate the amount of power coming from solar, each led corresponds to 20% of the total power</p> <hr/> <p>Note: In this figure the amount of power coming from solar is 60% (3 yellow leds)</p> <hr/>


5.4 Button purple

Color	Name	Description
	Update	<p>Update in progress. Device will reset when updating is completed</p>

5.5 Button light blue

Color	Name	Description
	Standby	Prism in standby, ready for charging

5.6 Others

Color	Name	Description
	No internet	<p>No internet connection. User input available only from front panel or with AP connection.</p> <hr/> <p>Note: only last led is purple, the other leds behave normally</p> <hr/>