OpenStrom
Topics and Structure of Technical Specification

The goal of the technical specification document is to have a basis to design the solution. The high level requirements of our solution are listed here: https://docs.google.com/document/d/1AJmYfzObi8I9YgZ2tjCaWn-a7VfgmVBiYcD5WFaKjvU

The technical specification will probably have between 15 and 30 pages and look similar (in terms of formatting and outline) to this document: http://files.opencompute.org/oc/public.php?service=files&t=4bb682624927b315b4f3d8a30f022c19

The document should be broken down into individual sections that outline the solution. This is not a complete list of aspects that need to be considered for each section of the document, but rather a starting point for you.

MCU:

- Is it powerful enough for our use cases? (cpu/ram)
    - What is the maximum sampling rate and how much data would need to be processed on the unit and how much data would be sent to the backend
    - Can we support energy disaggregation on the unit?
    - Can we support event detection (appliance turned on/off or unusual energy consumption patterns) on the unit?
    - Can we use data compression on the unit to reduce the amount of data we sent to the backend?
- What is available on the chip?
    - Wifi
    - LAN
    - GPIO (how many?)
    - I2C?
    - Other interesting functions (for future extensions and to allow users to hack the device with their own stuff)

Data / API:

- JSON and/or XML to transfer data (needs spec of format)
- Compression
- Frequency of sending data and packet size for different sampling rates
- Device should always initiate connection to backend, so that we can rule out any security issue by making the device accessible to the internet. So no incoming connections. This means that the switching of circuits needs to be triggered by the confirmation message after sending data or by some polling mechanism. Also consider “missed” messages or stuck relays, so the device should also receive the total expected relay status from the server, not only a single on/off command.

Sensors:

- Which data can be collected by sensors?
- What is the resolution of the sensor and what the maximum collection frequency?

Relays:

- Expected lifetime of relay
- Power draw of relay
- Plug-in/Shield for relays to save cost? Basically this would mean that the device works without relays, but we can just add a special relay board to add switching functionality. (using the same housing and pretty much the same manufacturing steps to get two versions of the device). What are the advantages/disadvantages of this approach?

Power Supply:

- Expected lifetime
- Temperature / heat

Housing/Enclosure:

- Do we need a custom mold? Or can we use a standard housing? Which one?
- Heat issues
- Electric magnetic shielding required?
- Wifi antenna required?

Firmware:

- How does the initial setup of the network connectivity work? (see http://www.ti.com/lit/wp/swry011/swry011.pdf). Should support: Wifi Protected Setup , Access Point Mode until internet connectivity is established, Smart Config app via phone
- How does the user/device enter the initial setup mode?
- How can users update the firmware? Can they brick the device? How to unbrick?
- How can the initial firmware be loaded during manufacturing (fast)
- How can we assign a unique device identifier that we can use to connect devices to the backend service
- Do we need to set MAC address during manufacturing?
- What security considerations do we need to consider?

Legal / Regulatory

- Which laws and regulations do we need to consider to market the device in the US and in the EU?

PCB:

- What connectors do we need? (how do they align with the housing?)
- Do we need a reset or on/off switch? What happens with the relays if we turn the device off?
- Do we support add-ons, shields or other extensions that users can use to add functionality to the device? (like additional sensors for e.g. temperature?)

Pricing:

- Are we at the €30-€35 BOM cost?

Components:
The component selection is part of the design phase, however the technical specification should already reference the different options listed in the requirements and do a quick assessment if these components are suitable. Use a table format that has the component and the relevant parameters at the end of each section.