1. ~~Имя протокола (сохраняем по 5 устройств в одном файле)~~
2. ~~Добавить поле номер прошивки в web конфигурацию~~
3. ~~Добавить адрес блока питания ЛБП в формате ip:port~~
4. ~~Добавить обработку исключений и метод get в файл read\_write\_eeprom.py добавить по три попытки считывания конфигурации.~~
5. ~~Добавить обработку исключений в файл формирования протокола и так же возвращать true/false при удачной и неудачной записи~~
6. Начать логику расписать старт устройства, считывание его конфигурации, замеры погрешностей – в общем первое включение
7. Обдумать заряд АКБ
8. Переделать обработку исключений ЛБП класс: power\_supply в devices.py

Нюанс: там, где требуется результат в виде данных создавать get методы, если результат в виде удачно/неудачно возвращать напрямую true/false.

FIRST BLOOD (Выводим протокол в любом случае)

Def first\_start

1. ~~Подключаем ЛБП~~
2. ~~Ждем включения~~
3. ~~Проверяем состояние устройства~~

Def configurate\_check

1. ~~Считываем версию прошивки~~
2. ~~Сохраняем версию прошивки~~
3. Передаём серийник в Log, передаём серийник в Web (опционально)
4. ~~Сохраняем серийник~~
5. ~~Считываем и сохраняем настройки питания~~
6. ~~Рассчитываем погрешности~~
7. ~~Подаём нагрузку~~
8. ~~Проверяем состояние устройства~~
9. ~~Считываем измерения и сравниваем.~~
10. ~~Подаем нагрузку max (по 5 A на канал)~~
11. ~~Считываем измерения и сравниваем.~~
12. ~~Проверяем состояние устройства~~
13. Проверяем IN1 u min
    1. Проверяем состояние устройства
    2. Проверяем IN1 u max
    3. Проверяем состояние устройства
    4. Проверяем IN1 u nom
    5. Отключаем IN1
    6. Проверяем состояние устройства
    7. Проверяем IN2 u min
    8. Проверяем состояние устройства
    9. Проверяем IN2 u max
    10. Проверяем состояние устройства
    11. Проверяем IN2 u nom
    12. Отключаем IN2
    13. Проверяем состояние устройства
14. Включаем IN1 и IN2 (БП) (итерация в 5 раз)
    1. Контроль пинга
    2. Проверяем состояние устройства
    3. Мгновенно отключаем IN1
    4. Проверяем состояние устройства
    5. Мгновенно отключаем IN2
    6. Проверяем состояние устройства
    7. Контроль пинга
    8. Проверяем состояние устройства
    9. Мгновенно включаем IN2
    10. Проверяем состояние устройства
    11. Мгновенно включаем IN1
    12. Добавляем данные в протокол

**(ВОЗМОЖНО В ПРОМЕЖУТКЕ МЕНЯЕМ УСТАВКИ 9:1 АНАЛОГИЧНАЯ ПРОВЕРКА С ПЕРЕГРУЗОМ)**

1. Делаем КЗ на OUT1
   1. Проверяем состояние устройства
   2. Снимаем КЗ с OUT1
   3. Проверяем состояние устройства
   4. Делаем КЗ на OUT2
   5. Проверяем состояние устройства
   6. Снимаем КЗ с OUT2
   7. Проверяем состояние устройства
   8. Добавляем данные в протокол
2. Включаем ТЭН.
   1. Проверяем состояние устройства
   2. Проверяем ТЭН и датчик
   3. Обрываем связь
   4. Проверяем состояние устройства
   5. Проверяем ТЭН и датчик
   6. Восстанавливаем связь
   7. Проверяем состояние устройства
   8. Проверяем ТЭН и датчик
   9. Отключаем ТЭН
   10. Добавляем данные в протокол
3. Проверяем состояние устройства
4. Отключаем БП и подключаем ЛБП
5. Проверяем состояние устройства
6. Считываем и проверяем измерения
7. Заносим в протокол
8. Отключаем ЛБП
9. Снимаем нагрузку
10. Делаем запись в файл протокола.

Дополнительно:

def measurements\_check(self):

- добавить расчет тока нагрузки (считывать параметр тока и давать до предела) реализовать циклом.

def check\_voltage\_thresholds(self):

добавить гистерезис учет гистерезиса (с учетом погрешности)

(проверить АКБ, проработать write\_eeprom)