Этот скрипт написан на языке Bash и предназначен для тестирования и калибровки временной карты SA53, работающей без внешнего осциллятора. Основная цель скрипта – проверка поведения временной карты при различных условиях, таких как потеря GPS-сигнала или изменение параметров калибровки.

**Основные функции скрипта:**

1. **read\_SA53\_int\_value()**: Читает целочисленное значение параметра с карты SA53 через Python-скрипт timetickler.py, который взаимодействует с картой.
2. **SA53\_start\_disc() и SA53\_stop\_disc()**: Включают и отключают дисциплинирование (процесс калибровки) временной карты. Также настраивают параметры, такие как PhaseLimit и PhaseMetering.
3. **SA53\_get\_phase()**: Читает текущее значение фазы с карты SA53.
4. **read\_SA53\_status() и timed\_read\_sa53\_status()**: Собирают и выводят ключевые параметры временной карты, такие как Disciplining, TauPps0, DigitalTuning, и записывают их в CSV-файл. Также функция timed\_read\_sa53\_status() периодически выполняет эту задачу в течение заданного времени.
5. **write\_csv\_header()**: Создает заголовок для CSV-файла, где будут храниться результаты тестов.
6. **sa53\_holdover\_test()**: Тестирует поведение временной карты в условиях "холдовер" (работа без входного сигнала, например, после потери GPS).

**Деградация и восстановление:**

Скрипт содержит два метода деградации и один метод восстановления:

1. **degrade\_sa53\_method1()**: Принудительно ухудшает калибровку, отключая дисциплинирование и задавая неверные значения DigitalTuning.
2. **degrade\_sa53\_method2()**: Использует небольшое значение Tau и большое значение CableDelay, чтобы намеренно калибровать карту на неправильные значения.
3. **fix\_degrade\_sa53\_method1()**: Восстанавливает корректную работу карты после деградации методом 1.

**Основной тест калибровки:**

Функция **converge\_test()** выполняет комплексное тестирование, включая деградацию, восстановление и проверку времени стабилизации при различных значениях Tau. В конце теста проверяется, насколько хорошо карта сохраняет точность в условиях холдовера.

**Как запустить:**

1. Убедитесь, что скрипт timetickler.py находится по указанному пути и корректно работает.
2. Сохраните код в файл, например, sa53\_test.sh.
3. Дайте файлу права на выполнение:

bash

Копировать код

chmod +x sa53\_test.sh

1. Запустите скрипт:

bash

Копировать код

./sa53\_test.sh

**Примечания:**

* Для работы скрипта требуется Python и дополнительные зависимости, используемые в timetickler.py.
* Скрипт создает и записывает данные в файл self\_holdover\_log.csv для последующего анализа.
* Можно настроить параметры тестов, такие как значения Tau, длительность холдовера и прочее.

Этот скрипт полезен для отладки и настройки работы временной карты SA53 в сложных условиях, включая потерю GPS-сигнала.