Программа контроля состояния фидерных линий комплексов громкоговорящего оповещения (ПК ФЛ КГО)

Описание функциональных характеристик программного обеспечения

2026

Содержание

Назначение

Программное обеспечение контроля состояния фидерных линий предназначено для обеспечения корректного функционирования усилительно-коммутационного блока Гром входящего в состав Комплекса громкоговорящего оповещения П-166 ИТК ОС КГО.

Основу ПО составляют три модуля системы stmmon (STMmonitor) для платформы RockChip RV1126, предназначенные для автоматизированного тестирования фидеров (акустических каналов) – определения их состояния (подключено, обрыв, короткое замыкание) на основе анализа записанного звукового сигнала.

Функциональные характеристики

Взаимодействие модулей

1. fidtest_Psound воспроизводит тестовый сигнал на фидер.
2. fidtest_Csound (в перспективе) записывает ответный сигнал с фидера в WAV-файл.
3. Внешний планировщик (или по команде) запускает stmmon_analiz_v1 над этим WAV-файлом.
4. Результат анализа (OK/SC/BR) отправляется по multicast для дальнейшей обработки.

1. stmmon_analiz_v1.c – модуль анализа WAV-файлов

Назначение:

Читает WAV-файл, полученный в результате записи тестового сигнала с фидера, выделяет амплитуды двух каналов (левый – токовый канал **I**, правый – канал напряжения **U**), применяет пороговые лимиты и принимает решение о состоянии фидера. Результат отправляется по multicast в формате JSON.

Основные функции:

- `main()` – точка входа:
- Инициализация отладки, лог-файла.
- Разбор аргументов командной строки (см. `ccnw_cmdline_processing`).
- Инициализация multicast-сокета.
- Проверка входного WAV-файла (`check_file`).

•Открытие входного и выходных файлов (текстовый .txt для детального лога и WAV с преобразованными данными).

•Чтение WAV-заголовка (44 байта), затем в цикле читаются сэмплы заданной разрядности (width = 4 для 16 бит, 8 для 32 бит).

•Для каждого сэмпла берётся модуль (абсолютное значение) и сравнивается с порогами L_t и R_t – если значение ниже порога, оно зануляется.

•Полученные массивы t32_L и t32_R накапливаются, также ведётся статистика (stat_count).

•По окончании чтения вызывается analiz(), затем результат отправляется через send_response().

•Завершается записью параметров и статистики в выходной текстовый файл.

•analiz(int32_t *L, int32_t *R) – ядро анализа:

•Пробегает по массивам от индекса skip до lim (задаётся окно контроля).

•Подсчитывает количество сэмплов в каждом канале, превышающих соответствующие лимиты (L_t и R_t) – получает counter_L и counter_R.

•Принимает решение:

•Если counter_R == 0 или меньше R_minlim/4 → возвращает FID_ST_SC (короткое замыкание).

•Если counter_L == 0 → FID_ST_BR (обрыв / не подключено).

•Если оба счётчика превышают минимальные временные лимиты (L_minlim, R_minlim) → FID_ST_OK.

•Иначе возвращает FID_ST_BR или -1 (неопределено).

•send_response() – формирует JSON-строку вида {"fiders":[{"P":"номер","S":"статус","L":counter_L,"R":counter_R}]} (для фидера 6 добавляется поле "tend":"end") и отправляет через multicast-сокет.

•stat_init(), stat_count(), stat_print() – механизм построения гистограммы распределения амплитуд для отладочных целей.

•ccnw_cmdline_processing() – парсинг аргументов командной строки: установка --skip, --lim, --L_t, --R_t, --L_minlim, --R_minlim, --width, --audiofile (номер фидера), --debug_on/off, --logging_to_file_on/off и др.

•check_file() – проверяет существование и размер WAV-файла; если размер равен 44 (пустой файл), пытается повторить проверку с задержкой (до 3 раз). Формирует имена выходных файлов.

•stmmon_analiz_init_ip_backend_multicast() – создаёт UDP-сокет для multicast-рассылки (адрес и порт по умолчанию из stmmon_storage.h).

2. fidtest_Csound.c – модуль потока записи звука

Назначение:

Реализует рабочий поток, который ожидает команды из FIFO (именованного канала) для управления записью звука. На данный момент код содержит только каркас: читает данные из FIFO и выводит их в отладку, реальная запись не реализована.

Основная функция:

•fidtest_thread_Csound(void *arg) – функция потока:

•Получает указатель на структуру pthread_params_t, содержащую контекст comm_pool_s.

•Извлекает дескриптор FIFO для чтения (fd_fifo_csound_rd).

•В цикле (пока активен глобальный стоп-маркер) вызывает poll() с таймаутом 500 мс.

•При появлении данных читает их в буфер размером 1024 байта и выводит отладочное сообщение.

•Обрабатывает ошибки poll() и чтения, ведёт статистику ошибок в структуре comm_pool_s.

•При обнаружении стоп-маркера или ошибок выходит из цикла и завершает поток.

3. fidtest_Psound.c – модуль потока воспроизведения звука

Назначение:

Поток, управляющий воспроизведением тестового сигнала на фидер через звуковую карту (ALSA). Поддерживает инициализацию устройства и воспроизведение заранее встроенного сигнала (из файла data_Testfile.c). Также способен принимать команды из FIFO.

Основные функции:

•fidtest_thread_Psound(void *arg) – основной поток:

•Аналогично потоку записи, получает контекст и дескриптор FIFO.

•Выполняет инициализацию ALSA (вызов playback_init()), хотя этот код находится внутри условного блока (if (0) – по умолчанию отключен).

•В цикле ожидает данные из FIFO через poll(), читает и выводит.

•В блоке инициализации (закомментирован) воспроизводит встроенный массив data_Testfile несколько раз через snd_pcm_writei().

•playback_init() – открывает устройство hw:1,0 (или default:CARD=SGTL5000audio), устанавливает параметры: частота 48000 Гц, 2 канала, формат S16_LE, готовит интерфейс к работе, выделяет буфер.

•writebuf(snd_pcm_t *handle, const void *buf, long len, size_t *frames) – вспомогательная функция для записи с повторными попытками при ошибках -EAGAIN, использует snd_pcm_recover() для восстановления.