1. Ф. NPD\_EXPORT int npd\_prepare(np\_handle \*h, int \*width);

Функция лишняя- подготавливать нечего. Пики сбрасывать нечем. Пики это результат

разгона двигателя. Во время разгона возникают переходные прцессы в сети. Сначала нужно дать движение - резогнать двигатель, а потом Дать команду на сьемку. Я сбросить

могу только помеховый заряд матриц, который я и так сбрасываю лишней сьемкой одной строки.

Пики это результат, того что детектор ничего не делает какие то время. Двигатель тут не причем.

Вы сами это установили, что если детектор не снимает продолжительное время, мы получаем пики. Сегодня нужно пики сбрасывать, завтра ещё что то, после завтра проверять какие то параметры. Это всё детали вашей реализации.

1. Размер снимка надо рассчитывать по количеству байт в снимке.

Один байт – величина не делимая. Канал оперирует не строками, а байтами. \*Если в канале

произошла ошибка и в принятой строке не будет хватать нескольких байт, то счетчик строк, все равно увеличится на единицу. И проверку работы канала все равно надо вести по количеству байт.

И вы должны проверять кол. возвращенных байт. Проверка по количеству возвращенных строк не

не гарантирует отсутствие ошибок т.к.\*

Количество байт передают в API Windows, и мы не должны изобретать велосипед Тем более что для Вас в программе это принципиально ничего не меняет.

Причем тут WINAPI, мы хотим абстрагироваться от реализации детектора. Мы Вам отправляем, сколько времени мы хотим снимать, Вы уже сами считаете, сколько байт вам нужно. Нас не интересуют Ваши формулу расчета байтов, даже если они очень простые сегодня, а завтра сложные. Нам нужна картинка, по заданному количеству времени. Мы понимаем, что по времени наш запрос может быть выполнен с погрешность, нас это устраивает.

3. В ф. NPD\_EXPORT npd\_image \*npd\_capture(np\_handle \*h, uint32\_t msec); Я опять вижу аргумент uint32\_t msec . Я уже обьяснял, что задавать размер снимка по времени-нельзя:

См. предидущее письмо.

1. Вместо Ф :

NPD\_EXPORT int npd\_prepare(np\_handle \*h, int \*width);

NPD\_EXPORT npd\_image npd\_capture(np\_handle \*h, uint32\_t msec);

Должна быть одна ф.

Arg: handle датчика, указатель на указатель буфера, количество байт для сьемки

NPD\_EXPORT DWORD npd\_get\_bufer(np\_handle, BYTE\*\* bufer , DWORD num)

Tсли вернула не ноль, то это кол. принятых байт

Если вернула ноль, то NPD\_EXPORT int npd\_last\_error(np\_handle \*h); вернет ошибку.

В будующем хотелось бы иметь комментарии к h файлам и изменения, которые там вводятся от версии к версии.

Мои комментарии:

1)

Пики это не результат простоя детектора:

- На моей программе никаких пиков нет, сколько бы детектор не простаивал.

- Отключи движение в своей программе и попробуй снимок - никаких полос не будет.

- В детекторе нет памяти более чем на две строки и хранить там 10-15 порченых строк – негде

2)

-Задаваемое время – это недостаточный параметр для задания пораметров изображения.

-Количество необходимых байт для формирования изображения - это и есть универсальная абстракция, подходящая для любого датчика.

Формула: NUM = NumPixInString\*NumStringInImage \* 2 будет верна всегда и никогда не изменится.

Нет никаких сложных алгоритмов и завтра они тоже никак не изменятся.

Эта формула будет верна для любого датчика во все времена. И количесство Байт от датчика требуемое - это единственная точная количественная мера, которая характерезует его работу.

Детектор это именно файл и с ним надо работать как с файлом. Тем более что UNIX так рассматривает все устройства, а Win. Все блочные устройства, а к символьным детектор из-за большого потока данных не отнесешь. Других устройств ни в UNIX, ни в Win не бывает и если мы начнем изобретать свои, то обязательно наткнемся на что-либо очень плохое и потеряем только время и силы.

Я не могу по времени сканирования рассчитать сколько байт. Вам нужно переслать. Точность такого расчета составит ±6строк при изображении в 4сек.и времени накопления 2мсек.

В температурном диапазоне +5°С - +35°С, это без учета ухода генератора со временем за год

Это никуда не годится.

Только вы знаете по количеству строк необходимых для изображения сколько байт вам нужно.