

Краткое описание как работать с Floader.

Предварительно по swire посылается остановка/сброс CPU - SwireWrite(0x006f, 0x20, 1).
Затем сразу производим процедуру Activate:

- передаем в цикле SwireWrite(0x0602, 0x05, 1) несколько ms (от 2-х).
- проверяем выполнение Activate, к примеру, чтением по swire какого известного регистра.

Возможно и без Activate, но надо остановить CPU, сбросить регистры по умолчанию и настроить адрес исполнения CPU как при страте. Сбросить CPU и выполнить Activate проще.

Далее загружаем бинарник в адрес 0x8000 - SwireWrite(0x8000, floader, sizeof(floader)).

После загрузки запускается CPU - SwireWrite(0x0602, 0x88, 1).

В бинарнике floader.bin по адресу 4 расположен указатель на адрес структуры обмена командами EXT.

Далее общаемся с работающим floader по swire через эту структуру EXT в памяти TLSR8269.

Код (C):

```
1. typedef struct {
2.     volatile u32 faddr; // адрес Flash или другого устройства
3.     volatile u32 pbuf;  // адрес буфера в памяти TLSR
4.     volatile u16 count; // кол-во байт обработки
5.     volatile u16 cmd;   // команда Floader-y
6.     volatile u16 iack;  // оповещение о смене команды по swire
7.     volatile u16 oack;  // оповещение Floader-a о исполнении команды
8. } sext;
9. sext ext;
```

По старту Floader-a данная структура заполняется:

- ETX.faddr = Flash ID
- ETX.pbuf = адрес буфера в памяти TLSR
- ETX.count = максимальный размер буфера
- ETX.cmd = команда чтения ID Flash
- ETX.iack = номер версии Floader
- ETX.oack = 0

После старта Floader, через задержку в пару ms, по swire считываем EXT.
Проверяем размер ETX.count, FlashID, адрес буфера, номер версии Floader...

Если хотим подать команду чтения Flash, то изменяем:

1. ETX.faddr = адрес чтения Flash,
2. ETX.count = сколько хотим прочитать за раз (см. ограничение буфера)
3. ETX.cmd = FLASH_READ_CMD
4. ETX.iack увеличиваем на 1

ETX.pbuf не меняем (будем менять - это надо только при хитрых мультизадачах).

Передаем структуру ETX по Swire за раз, но не всю, без последних 2-х байт ETX.oack.

В цикле читаем по swire слово ETX.oack и ждем когда оно изменится от прошлого (изменяется на +1).

Это значит floader выполнил команду.

Читаем по swire буфер по адресу ETX.pbuf и размером ETX.count.
Всё – заданный кусок Flash прочитан.

Аналогично и с другими командами.

При записи блока в Flash предварительно, перед подачей команды в ЕХТ загружаем буфер по swire. Даем команду записи и можем грузить другой участок буфера по swire, пока CPU записывает предыдущий буфер.

Стирание сектора так-же можно расслоить по времени. Даем команду стирания сектора в ЕХТ, а пока CPU ожидает стирания заполняем буфер для записи. Заполнив буфер, проверяем выполнение команды стирания (ETX.oack = старое значение +1) и подаем команду записи буфера в сектор Flash.