



**Особенности разработки встраиваемого ПО для крейта LTR**

**Москва. Август 2009 г.**

## **ЗАО «Л-КАРД»**,

117105, г. Москва, Варшавское шоссе, д. 5, корп. 4, стр. 2.

тел. (495) 785-95-25

факс (495) 785-95-14

### **Адреса в Интернет:**

WWW: [www.lcard.ru](http://www.lcard.ru)

FTP: [ftp.lcard.ru](ftp://ftp.lcard.ru)

### **E-Mail:**

Общие вопросы: [lcard@lcard.ru](mailto:lcard@lcard.ru)

Отдел продаж: [sale@lcard.ru](mailto:sale@lcard.ru)

Техническая поддержка: [support@lcard.ru](mailto:support@lcard.ru)

Отдел кадров: [job@lcard.ru](mailto:job@lcard.ru)

### **Представители в регионах:**

Украина:	“ХОЛИТ Дэйта Системс, Лтд”	<a href="http://www.holit.com.ua">www.holit.com.ua</a>	380 (44) 241-67-54
Санкт-Петербург:	ЗАО “АВТЭКС Санкт-Петербург”	<a href="http://www.autex.spb.ru">www.autex.spb.ru</a>	(812) 567-7202
Санкт-Петербург:	Компания "Ниеншанц-Автоматика"	<a href="http://www.nnz-ipc.ru">www.nnz-ipc.ru</a>	(812) 326-59-24
Новосибирск:	ООО “Сектор Т”	<a href="http://www.sector-t.ru">www.sector-t.ru</a>	(3832) 22-76-20
Екатеринбург:	Группа Компаний АСК	<a href="http://www.ask.ru">www.ask.ru</a>	(343) 371-44-44
Екатеринбург:	ООО “Авеон”	<a href="http://www.aveon.ru">www.aveon.ru</a>	(343) 381-75-75
Казань:	ООО “Шатл”	<a href="mailto:shuttle@kai.ru">shuttle@kai.ru</a>	(8432) 38-16-00
Самара:	"АСУ-Самара"	<a href="mailto:prosoft-s@jiguli.ru">prosoft-s@jiguli.ru</a>	(846)-998-29-01

**Особенности разработки встраиваемого ПО для крейта LTR**

© **Copyright 1989–2007, ЗАО “Л-Кард”**. Все права защищены.

## Разработка программного обеспечения для крейта LTR.

Крейты LTR поставляются со встроенным в них программным обеспечением, позволяющим работать через интерфейсы USB и Ethernet. Однако для некоторых - критичных по времени реакции - приложений может возникнуть необходимость во встраивании управляющих алгоритмов в ПО крейта. В ПО крейта предусмотрена поддержка такой возможности.

Общий принцип работы выглядит следующим образом: с хоста производится настройка и запуск модулей, а ПО крейта выполняет требуемую обработку принимаемых от модулей ввода данных и, при необходимости, передает результаты обработки в хост и/или в модули вывода.

Чтобы создать свою программу для крейта необходимо загрузить (с сайта [www.lcard.ru](http://www.lcard.ru)) и модифицировать исходные файлы программы крейта LTR. Для сборки проекта должен быть установлен пакет VisualDSP++ версии 5.0 или выше.

Группа проектов крейта состоит из двух проектов:

1. HAL\_BF537 – библиотека для работы с периферийными устройствами микроконтроллера.
2. ltr030-bf-fos – рабочая программа крейта.

Предполагается, что проект расположен в каталоге [C:\prj](#). При расположении проекта по другому пути необходимо в опциях проекта указать пути включаемых файлов относительно нового каталога. Структура каталогов должна быть сохранена.

Проекты ПО крейта содержат следующие файлы и каталоги (относительно [C:\prj](#)):

include\extint.h – определение типов данных фиксированного размера.

lib\EMAC\_BF – библиотека для работы с EMAC-контроллером.

lib\FreeRTOS – операционная система реального времени FreeRTOS.org.

lib\lwip – TCP/IP стек.

ltr\ltr030\ltr030-bf-fos\HAL\_BF53X – библиотека для работы с периферийными устройствами микроконтроллера.

ltr\ltr030\ltr030-bf-fos\HARDWARE – библиотека для работы с подключенными к микроконтроллеру устройствами.

ltr\ltr030\ltr030-bf-fos\datahook.c – анализ принятых данных от модулей и вызов заданных пользователем обработчиков.

ltr\ltr030\ltr030-bf-fos\hook\_example.c – пример написания обработчиков принимаемых от модулей данных.

ltr\ltr030\ltr030-bf-fos\ProcessComand.c – обработка принятых по USB команд.

ltr\ltr030\ltr030-bf-fos\ProcessData.c – обеспечение обмена данными между модулями и хостом USB.

ltr\ltr030\ltr030-bf-fos\ethernet.c – обеспечение работы крейта через Ethernet.

ltr\ltr030\ltr030-bf-fos\main.c – управление крейтом.

Чтобы написать свои обработчики данных необходимо в файле GlobDefs.h определить константу DATA\_HOOK\_EN равной 1:

```
#define DATA_HOOK_EN (1)
```

Поддержка написанной пользователями обработки принимаемых от модулей данных обеспечивается в файле datahook.c. Алгоритм работы выглядит следующим образом: принимаемые от модулей данные (только данные — не команды) копятся в буферах (для каждого модуля свой буфер) и, после получения заданного количества данных, вызываются определенные пользователем обработчики.

Для задания пользовательских обработчиков используется массив `g_datahook_cfg[ N_CRATE_MODULES ]` структур `struct datahook_cfg`.

Структура `struct datahook_cfg` определена в файле `datahook.h`:

```
struct datahook_cfg
{
    /* Тип модуля */
    enum e_Module module;
    /* Размер принимаемых данных для вызова обработчика */
    int data_size;
    /* Обработчик принятых данных
```

