

1 Overview

This CodeKit software contains code to initialize the Élan™SC520 microcontroller from a system reset and prepare the processor to boot a Linux kernel from Flash memory. The code is derived from the previous SDRAM sizing/startup CodeKits. This CodeKit software is designed to be programmed into a Flash memory or ROM device at the Reset vector for use as system startup code. This CodeKit is complemented by the Linux Flash Kernel CodeKit (CK005900).

The main difference from the SDRAM Size CodeKit is that this CodeKit looks for and loads a Linux kernel rather than a Embedded C Library application. To accommodate this, this CodeKit adds crude support for the INT15 BIOS service call. This CodeKit will not support the use of a video card or an IDE hard drive. The General Software BIOS (CK005000) should be used instead if either of these is needed.

IMPORTANT: By loading this software or any portion thereof, you agree to all the terms of the License Agreement in the file LICENSE.PDF. Do not use this software until you have carefully read and agreed to the terms and conditions in the Agreement. If you do not agree to the terms of the Agreement, do not use this software or any portion thereof.

For support: e-mail epd.support@amd.com
 or call 1-800-222-9323

1.1 Tool and System Requirements

The following target hardware and tools are required to build and execute this CodeKit software.

ITEM	DESCRIPTION
Target Hardware	ÉlanSC520 microcontroller customer development platform (CDP), revision 1.3, or the NetSC520 demonstration board
Monitor	Remon CK002803
Workstation Operating System	DOS 6.2, or Windows® 3.1, 95, 98, or NT
Tool Suite	Microsoft MASM compatibility driver version 6.110
Compiler	Microsoft C/C++ optimizing compiler version 8.00c
Assembler**	Microsoft MASM compatibility driver version 6.11
Linker*	Microsoft 16 bit linker Version 5.31.009 (included with MASM)

* Note the linker is part of the MASM 6.11 tool suite.

** Note MASM 6.11 is a separate product and is available with the Microsoft Developer Network on Disk 5 of the Office Test Platform and Development Tools Pack.

1.2 Installing the CodeKit Software

This CodeKit software is delivered as a self-extracting executable file. Double click on the EXE file to create a directory called C:\AMD\CK006000. The CK006000 directory contains the entire contents of the Linux Startup CodeKit.

1.3 Files and Directories

The EXE file contains the following files in a single directory.

FILE	DESCRIPTION
BUILDIT.BAT	A batch file that builds CodeKit software
CLEANIT.BAT	A batch file that cleans up intermediate files after a build
ENVIRON.BAT	A batch file that sets up the build environment
FLASHFS.BIN	A Linux file system image to be downloaded to flash
LICENSE.PDF	Licensing agreement for the software contained in the CodeKit
LOADLIN.CMD	A Remon command file for loading the Linux files into Flash memory on the NetSC520 demonstration board
LOADLINUX.BAT	A batch file that automates loading of all files on the NetSC520 board
LOADNET.CMD	A Remon command file for loading the startup code into Flash memory on the NetSC520 demonstration board
LOADNET2.CMD	A Remon command file for loading the startup code into Flash memory on the NetSC520 demonstration board
LSTART16.ABS	A binary image of the startup code that can be loaded into a Flash memory or ROM
LSTART16.ASM	The Linux Startup assembly language source example code
MAKEABS.C	Source file for the MAKEABS.EXE locator program
MAKEABS.EXE	An executable file that converts an EXE program into a binary ABS file for loading to Flash memory using a PROM programmer or emulator
MAKEFILE	A makefile that can be used instead of BUILDIT.BAT
README.PDF	This document in Adobe Acrobat PDF
SIZER.ASM	The SDRAM size/setup assembly language source example code
SMARTFLASH.EXE	A local copy of the SmartFlash utility for Remon; from CK003703
ZFI.BIN	The Linux kernel from CK005900

1.4 Setting Up the Build Environment

Before the CodeKit software is built, you must set up the build environment. To do this, run the ENVIRON. BAT command file. Running the command file sets up the environment variables listed in the table below.

This batch file assumes that the tools have been installed in their default locations on the C: drive.

ENV VARIABLE	SETTING
TOOLROOTDIR	C:\MSVC
INCLUDE	C:\MSVC\INCLUDE
LIB	C:\MSVC\LIB

1.5 Setting Up the Target Hardware

This section describes how to set up the target hardware when running the demonstration under Remon.

1.6 Loading and Running with Remon on the NetSC520

To load into Flash on the NetSC520, first install the Remon CodeKit. Open a DOS window and change directories to the CK006000 directory. Run the ENVIRON.BAT file, then load the files into the device by entering the command **loadlinux**. This DOS batch file will start Remon and load the LSTART16.ABS file to offsets FF800h and FFF800h in the Flash memory. It also places the Linux kernel, ZFI.BIN, at 0 in Flash memory, and the Linux file system, FLASHFS.BIN, at 100000h in Flash memory. When the command is finished executing, enter **z** followed by **g**, or press reset. The Linux Startup Codekit will execute. LSTART16 will load the Linux kernel, which boots up the system. Connect a terminal emulator (set for 9600 bits/s, N, 8, 1) to J6 and you will see the Linux kernel console output.

The Linux kernel is a 2.2.18 kernel tailored for the NetSC520. It uses the new experimental Journaling Flash File System (JFFS). It will use a serial port console on port J6 at 9600 bits/s. No disk or video display are required. The kernel has a working Ethernet and TCP/IP setup. See CK005900 for more details.

The board will boot to a login prompt in about 30 seconds. Use “root” as the username and “password” for the password. You are free to add users and change passwords.

NOTE: The file system has a file in /etc/rc.d/init.d called S20network. You **MUST** edit this file to change the network parameters, especially the IP address, gateway, and broadcast address. Please do this before connecting to the network.

You can telnet to the board, or you can access the web page by typing <http://xxx.xxx.xxx.xxx>, substituting the IP address you have assigned to the board.

NOTE: Loading the Linux Startup CodeKit into the NetSC520 Flash memory will erase any prior contents of this Flash memory device. This Flash memory contains the General Software Embedded BIOS used to provide DOS capability on the NetSC520 board. If you need to return to using the General Software BIOS at a later time, it can be reloaded using the instructions in CK005000. Note also that the General Software BIOS on the NetSC520 has an option to boot the linux kernel that is loaded by this CodeKit. It could be used in lieu of this startup code, and is required if a VGA console or IDE hard drive are needed.

NOTE: Both the Linux Startup and the Linux Flash Kernel CodeKits must be rebuilt to use them on a CDP instead of a NetSC520. For this Linux Startup CodeKit, comment out the **NetSC520 equ 1** line at the beginning of lstart16.asm and rebuild the CodeKit. Follow the instructions in CK006000 to rebuild the kernel. The Remon cmd files will need slight modification to load the files to a CDP.

1.7 Building the CodeKit Software

Building this CodeKit software is simple:

1. Execute the ENVIRON.BAT file to set up your environment.
2. Execute the BUILDDIT.BAT file.

This builds the LSTART16.ABS file. This file can then be loaded to flash.

NOTE: Buildit also builds the MAKEABS.EXE application, which is used to convert LSTART16.EXE into a binary ABS file.

For help with MAKEABS.EXE, run MAKEABS with no arguments.

2 About the Software

LSTART16.ASM is a simple 16-bit mode assembly language startup program. It will automatically detect the type and size of SDRAM in the system, and configure the SDRAM controller accordingly. It will then copy a Linux kernel image from the start of Flash memory to location 7C00h in SDRAM and then jump to 7C00h.

2.1 LSTART16.ASM

LSTART16 is an x86 assembly program. It is written for use with Microsoft MASM 6.11.

This program includes a reset vector for the ÉlanSC520 microcontroller's Am5_x86® CPU. After the reset vector, this program will switch to big real mode and initialize some of the CDP resources such as the GP Bus. This program will then initialize the SDRAM and transition back to real mode.

LSTART16.ASM calls the routine in SIZER.ASM to set up the SDRAM controller.

LSTART16.ASM uses ORG statements to locate the descriptor tables and the reset vector relative to the start of the file. If the ORG positions are changed (for instance if code was added so that it overlapped following sections), care should be taken to correctly arrange all ORGs and the address constant in the global descriptor table. In addition, STARTABS.C will need to be modified by changing the second and third parameters in the call to CreateFile()

2.2 SIZER.ASM

SIZER.ASM contains the SDRAM sizing algorithm and also sets the SDRAM controller based on the results of the sizing algorithm. The program determines the number of column bits, the number of internal banks, and the ending address of the SDRAM DIMMS. This information is used to set the SDRAM controller registers.

The program expects a 32-bit real operating environment. Because no read/write memory exists in the system before this routine is called, it must be jumped to and the return address is placed in the BP register.

2.3 MAKEABS.C

MAKEABS.C performs a simplistic conversion of an EXE file into an ABS file, or ROM image, suitable for burning Flash memory on a programmer. This file reads a 16-bit DOS EXE file, removes the EXE header, and performs any necessary address relocations. This file is currently set to output a file that is intended to be fixed (by the PROM programming process) at FFFFF800h in the ÉlanSC520 microcontroller address space, or at the top of the BOOTCS ROM device. This location can be modified by changing the second and third parameters in the call to CreateFile(). When performing such a change, be sure that the file to be converted is also appropriately modified.

Note that more information about the code is supplied in the inline documentation.

Trademarks

© 2001 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD logo, combinations thereof, Élan, and Am5_x86 are trademarks, and Am5_x86 is a registered trademark of Advanced Micro Devices, Inc.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.