Readme File: Version 1.0 – Sept 27, 1999

# 1 Overview

This CodeKit software contains drivers for Microwire® EEPROMs and serial peripheral interface (SPI) devices found on the ÉlanSC520 microcontroller customer development platform (CDP). A small application is included that uses the drivers to read and write from the Microwire AT93C46A, and SPI251281EEPROMS on the CDP and to perform a direct loopback of the Data Out (DO) and Data In (DI) pins on the ÉlanSC520 microcontroller SSI interface.

**MPORTANT:** 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 | |
|---|---|
| Target Hardware | ÉlanSC520 microcontroller, revision 1.0 and above CDP |
| Workstation Operating System | DOS 6.2, or Windows® 3.1, 95, 98, or NT |
| Tool Suite | Microsoft® Visual C++ V1.52 |
| Compiler* | Microsoft C/C++ optimizing compiler version 8.00c |
| Assembler** | Microsoft MASM compatibility driver version 6.11 |
| Linker* | Microsoft segmented executable linker version 5.60.339 Dec 5 1994 |

* Note the compiler and linker are part of the Visual C++ V1.52 tool suite.
** Note MASM6.11 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\CK003300. The CK003300 directory contains the entire contents of the SSI driver CodeKit.

## *1.3 Files and Directories*

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

| FILE | DESCRIPTION |
|---|---|
| ASP_SSI.BAT | Batch file that calls SSI executable to perform Microwire, SPI, and SSI loop-back tests. |
| ASPENCLO.H | 32k clock function header file. |
| ASPENRTC.H | RTC utility function header file. |
| ASPENRTC_COMMON.C | This file contains some common utilities for the ÉlanSC520 microcontroller RTC tests. |
| ASPENSSI.C | ÉlanSC520 microcontroller SSI driver code. |
| ASPENSSI.H | This file contains hardware-specific declarations for the ÉlanSC520 micro-controller. |
| BUILDIT.BAT | Builds the CodeKit executable. |
| CLEANIT.BAT | Cleans intermediate build files. |
| ENVIRON.BAT | A batch file that sets up the build environment. |
| IO32BIT.ASM | Code for 32-bit I/O. |
| IO32BIT.H | Prototypes for 32-bit I/O functions. |
| LICENSE.PDF | Licensing agreement for the software contained in the CodeKit. |
| SSI.EXE | SSI driver executable. |
| README.FM | This document in Adobe Framemaker format. |
| README.PDF | This document in Adobe Acrobat PDF format. |
| STDINCS.H | A header file with standard typedefs and structures. |
| UTILITIES.C | Code for writing/reading memory-mapped configuration registers. (MMCR). |
| UTILITY.H | Contains some useful operations and data members. |

### *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.

| ENVIRONMENT VARIABLE | SETTING |
|---|---|
| TOOLROOTDIR | C:\MSVC |
| INCLUDE | C:\MSVC\INCLUDE |
| LIB | C:\MSVC\LIB |

### *1.5 Setting Up the Target Hardware*

You must place a jumper connecting pins 3 and 4 on JP17 on the CDP. This enables the direct loopback of the SSI from the Data Out (DO) to the Data In (DI) pins. No other setup is necessary.

### *1.6 Loading the Code*

The executable is a DOS real mode application so it can not be run for a DOS box, it needs to be run from real DOS.

## 2 About the Software

The demonstration application does three things:

- Performs a direct loopback of the SSI from the DO pins to the DI pins on the SSI interface and reports any failures.
- Reads and writes to the Microwire AT93C46A EEPROM and reports any failures.
- Reads and writes to the SPI AT2512 EEPROM and reports any failures.

To run the demonstration application, run the ASP_SSI.BAT file included in the CodeKit. This calls the SSI.EXE file with the arguments to run the three tests.

The SSI.EXE file takes four command line arguments.

SSI <arg1> <arg2> <arg3> <arg4>

#### 2.0.1 SSI Clock Divisor <arg1>

0 = 8 MHz

1 = 4 MHz

2 = 2 MHz

3 = 1 MHz

4 = 512 kHz

5 = 256 kHz

6 = 128 kHz

7 = 64 kHz

### 2.0.2 Array Index for Location from Which Data Begins Transmitting <arg2>

The array index is not particularly meaningful for this application. Just set this option to 0. This field controls where in the transmit array the data starts transmitting, and when the data is placed into the receive array.

### 2.0.3 Clock and MSB Combination Control <arg3>

This parameter controls three of the SSI features:

1. Whether or not to invert the SSI transmit clock
2. Transmitting and receiving MSB first or LSB first
3. Whether or not to invert the phase of the data

The table below shows all the possible combinations and the argument for each combination.

| Value | Invert Clock | Invert Phase | MSB First |
|---|---|---|---|
| 0 | No | No | No |
| 1 | No | No | Yes |
| 2 | No | Yes | No |
| 3 | No | Yes | Yes |
| 4 | Yes | No | No |
| 5 | Yes | No | Yes |
| 6 | Yes | Yes | No |
| 7 | Yes | Yes | Yes |

### 2.0.4 SSI Command <arg4>

1 = Microwire 8 bit

2 = Receive data

3 = Simultaneous transmit and receive (loopback mode)

4 = SPI transmit

5 = Microwire 16-bit

Using the four arguments supplied, you can create numerous combinations of SSI transactions from the command line, but most of the transactions are reads and writes to the EEPROMs to make sure they are functioning. You have to modify the code to perform actions such as erasing or completely reprogramming the EEPROMs.

The SPI test portion of the CodeKit is currently programmed to write and read exactly three bytes to the SPI EEPROM. The Microwire test writes and reads back ten bytes from the Microwire EEPROM. The loopback test sends one byte out on the DO pin of the ÉlanSC520 microcontroller SSI interface and receives the data on the DI pin. You can change the number of bytes transmitted and read for each of these tests in the ASPENSSI.C file.

## *2.1 Functions of Interest*

**MAIN()** This function performs all the command line processing for the SSI.EXE program and can most likely be deleted if you only need one type of SSI function from this CodeKit. MAIN() does provide good examples of using the supplied driver functions to write to different kinds of serial EEPROM devices.

**SIMULTANEOUS_TRANSMIT_RECEIVE()** This function shows how to simultaneously transmit and receive data using the DO and DI pins (these are pins 3 and 4 on JP17) on the ÉlanSC520 microcontroller SSI interface.

**MODIFYMW()** Call this function to perform changes to the Microwire EEPROM on the ÉlanSC520 microcontroller CDP. MODIFYMW() also handles calls to set up PIOs and chip selects on the CDP to select the Microwire AT93C46A EEPROM.

**READMW()** Call this function to read the contents of the Microwire EEPROM on the ÉlanSC520 microcontroller CDP.

**MODIFYSPI()** Call this function to modify the SPI AT2512 EEPROM on the ÉlanSC520 microcontroller CDP. MODIFYSPI() contains code to program the Super I/O chip to select this EEPROM for writing to.

**READSPI()** Call this function to read the contents of the SPI AT2512 EEPROM.

**NOTE:** Remember that the SSI chip selects (to JP17), the Microwire chip select, and the SPI chip selects are all driven through the Super I/O chip on the ÉlanSC520 microcontroller CDP. Keep this in mind when making changes to the code.

Please look at the in-line documentation for more programming information.