

AX8052 Serial Boot Loader



ON Semiconductor®

www.onsemi.com

APPLICATION NOTE

Introduction

Many products need the ability to update the firmware in the field. A solution to this requirement is the use of a boot loader, a small program located in the microcontroller firmware that communicates with an external entity, such as a technician's personal computer. The boot loader, when activated, receives an updated application firmware and stores it into the microcontroller flash. On next restart of the microcontroller, the updated firmware receives control.

This application note describes a simple boot loader that communicates to the external entity using the serial port UART0. It is intended as a building block for user's special purpose boot loaders.

The boot loader communicates using 38400 Baud, 8 Bits, 1 Stop bit and no parity bits. The transmit signal is output on PB4, receive signal is expected on PB5. Upon reset, the boot loader takes control. If Switch 5 is not pressed and an application firmware has been stored in the flash, it passes control immediately to the application firmware. If no application firmware is stored or Switch 5 is pressed, the boot loader starts and outputs a banner. A simple protocol is then used to communicate with the boot loader:

Command	Description
?	Output Banner
K	Erase ("Kill") Application
Z	Check whether the Application Area is Erased ("Zeroed")
:	Start of an Intel Hex file line. The line is parsed and written into the Application Area. The Application Area must have been erased before.
R	Run Application
V	VDDIO Measurement

A typical Boot Loader Session would therefore look like:

1. "?" – to check communication with the boot loader
2. "K" – to erase the application area
3. Upload the .hex file produced by the (SDCC) compiler
4. "R" – to run the application

Code Overview

Memory Layout

In order to minimize the constraints on the application, the boot loader is placed at the end of the memory, just before the calibration sector. The only constraint on the application code is that it may not overflow into the boot loader code, and that it must place a jump instruction at address 0, the reset vector.

The memory layout looks like this:

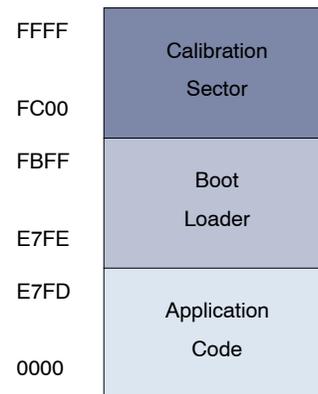


Figure 1.

Interrupt Vector Table

In order to avoid interrupt overhead for the application, the boot loader does not use any interrupts. It operates the UART in a polled fashion.

The boot loader however has to intercept the reset vector. The boot loader prevents writes to addresses 0x0000–0x0002; instead, it extracts the target address and writes it to the address 0xE7FE. The boot loader stores its own reset vector into addresses 0x0000–0x0002.

After reset, the microcontroller thus jumps into the boot loader code. The boot loader then examines Switch 5, and the address stored at address 0xE7FE, and then decides whether to start the application. To start the application, it just jumps to the address extracted from address 0xE7FE.

Care has to be taken however when erasing page 0 (where the interrupt vector table is located) to minimize the window where a power interrupt could render the microcontroller in a state where the boot loader does not start. To achieve this, the application area is erased like this:

1. Erase Page 1
2. Write a Jump to the Boot Loader reset vector at the beginning of Page 1
3. Erase Page 0
4. Write the Boot Loader Reset Vector at Address 0x0000
5. Erase All Application Pages except Page 0

After each of the above steps, the microcontroller is in a bootable state. For example, if it is reset after step 3, it still boots, thanks to the Jump in Page 1, after executing 1024 times MOV R7,A, which is harmless.

As an extra precaution, the boot loader checks VDDIO is above a configured threshold (BOOTLDR_MINVDD) before starting to erase.

UART

Factory calibrated AX8052 devices should be able to use the fast RC oscillator for baud rate generation. The example code can also be configured (BOOTLDR_CALFRCOSC) to turn on FRC oscillator calibration against a 32kHz watch crystal.

IDE Project

This Application Note ships a project file for Axsem's AX8052IDE. Version 1.5 or later is needed to compile the Boot Loader.

Since the Boot Loader requires a non-standard interrupt vector table, the project ships modified C startup code contained in the file crcstart.asm. In order to be able to assemble this file, the project needs to know the installation path of the assembler. The assembler path can be set in Project→Build Options...→bootloader→Custom Variables, variable assembler should point to the installation directory of sdas8051.exe (usually C:\SDCC\bin\sdas8051.exe)

If the boot loader starting address should be changed (for example because more space is needed for the boot loader),

it must be changed in two places. Project→Build Options...→bootloader→Linker Settings→Other Linker Options lists the option “-Wl-bHOME=0xe800”, which specifies the starting address of the boot loader. Furthermore, the address of the variable “jmp_vector” should be adjusted to lie 2 bytes below “HOME”.

Conclusion

This application note presented an example Boot Loader intended as a building block for Customer Boot Loaders. The beauty of this concept is that the same application binary / hex file can be used with or without the Boot Loader. It does not cause any interrupt overhead.

There are however a few areas where improvements may be worthwhile:

- The method for activating the Boot Loader, pressing SW5 during reset, works on the DVK-2 hardware, but probably not on customer hardware
- The protocol should be changed to allow checking the integrity of the downloaded code
- While communicating with the Boot Loader using a generic terminal program like HyperTerm or Minicom is simple for demonstration purposes, a Personal Computer Program should be written that handles Firmware Upgrades in the field without user interaction.

ON Semiconductor and the  are registered trademarks of Semiconductor Components Industries, LLC (SCILLC) or its subsidiaries in the United States and/or other countries. SCILLC owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of SCILLC's product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent-Marking.pdf. SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:

Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
Phone: 303-675-2175 or 800-344-3860 Toll Free USA/Canada
Fax: 303-675-2176 or 800-344-3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800-282-9855 Toll Free
USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910
Japan Customer Focus Center
Phone: 81-3-5817-1050

ON Semiconductor Website: www.onsemi.com

Order Literature: <http://www.onsemi.com/orderlit>

For additional information, please contact your local Sales Representative