Embedded System Design Techniques™

Bootloader Design for MCUs

Session 5: Troubleshooting Techniques

January 29th, 2016

Jacob Beningo, CSDP

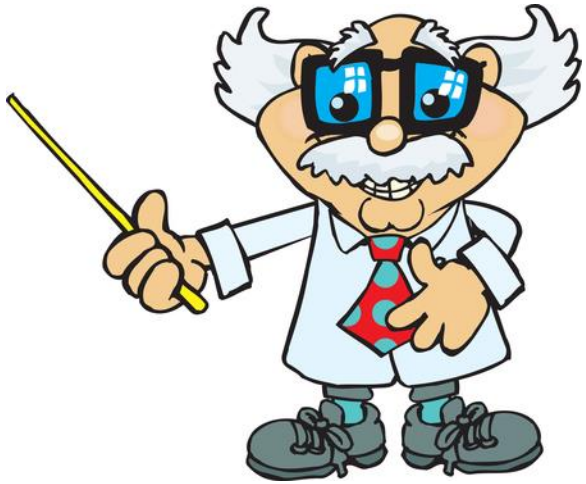
DesignNews

CEC CONTINUING EDUCATION CENTER

Presented by: Digi-Key ELECTRONICS

Session Overview

- Debugging Bootloaders
- Issues with Flash
- Image Verification
- Valid Reset Vectors
- Test Cases



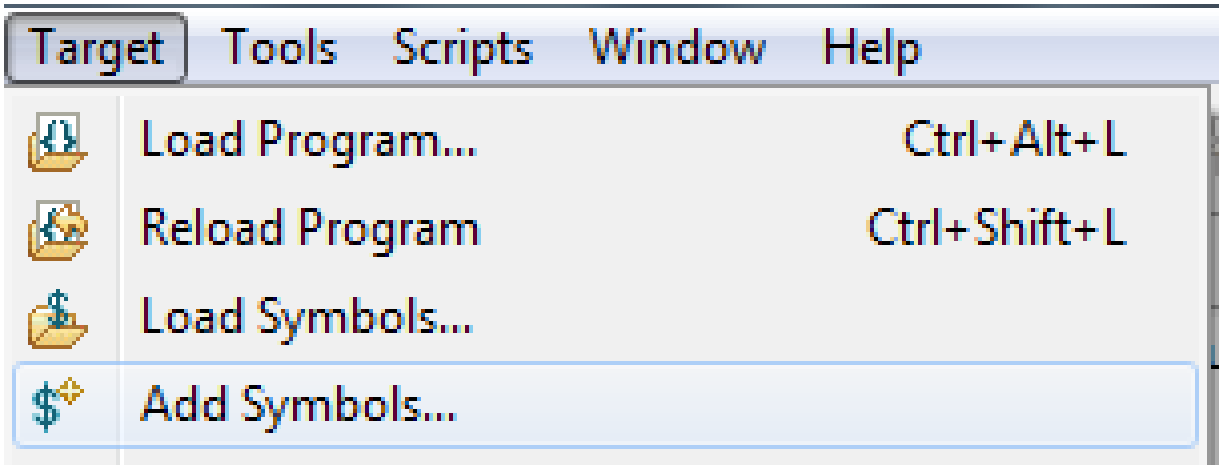
Debugging Bootloaders

- What could possibly go wrong?

Debugging Two Applications

Steps to debug the application:

- Run the boot-loader in debug
- Flash the application image onto the system
- Reset the processor
- Add symbols from the IDE and select the application symbols



Issues with Flash

- Flash controller NOT initialized properly
  - Clock rate
  - Clock gating
- Write method
  - Byte or Page?
- Checksum
  - Wrong place
  - Written backwards
  - Not written!

Resolutions for Flash

- Write a test harness to verify flash settings and write to various locations in flash
- Flash driver should
  - Write record to flash
  - Verify write by reading back and comparing
  - Failure to write should generate an error!
- Use example code for flash as a jumping off point for your own driver

Image Verification

- Export the flash image from the IDE while in debug mode running only the application
- Export the flash image from the IDE while in debug mode running the boot-loader with the application
- Use WinMerge to compare the images to determine if there is a difference between the image when loaded through the boot-loader

Valid Reset Vector?

- One of the major causes of staying in the bootloader

```
// If BootConfig Byte is set, application does not exist, or the calculated application
// checksum does not equal the stored application checksum, attempt to load hex file.
if(Flash_BootEnabled() ||
   Flash_AppVector(PROGRAM_FLASH_BASE) == 0xFFFFFFFF ||
   Flash_AppVector(PROGRAM_FLASH_BASE) == 0x00000000 ||
   Flash_GetChecksum(APPLICATION_CHECKSUM) != Flash_CalcChecksum(PROGRAM_FLASH_BASE, PROGRAM_MEM_STOP)
  )
{
  // Check to see if the application was corrupted. If so check the backup application and if it is okay then restore it.
  if(Flash_AppVector(PROGRAM_FLASH_BASE) == 0xFFFFFFFF ||
     Flash_AppVector(PROGRAM_FLASH_BASE) == 0x00000000 ||
     Flash_GetChecksum(APPLICATION_CHECKSUM) != Flash_CalcChecksum(PROGRAM_FLASH_BASE, PROGRAM_MEM_STOP)
    )
  {
      // The application was corrupt or incomplete, check the back-up application
      if(Flash_AppVector(PROGRAM_BACKUP_FLASH_BASE) == 0xFFFFFFFF ||
         Flash_AppVector(PROGRAM_BACKUP_FLASH_BASE) == 0x00000000 ||
         Flash_GetChecksum(APPLICATION_BUFFER_CHECKSUM) == Flash_CalcChecksum(PROGRAM_BACKUP_FLASH_BASE, PROGRAM_BACKUP_FLASH_END)
        )
      {
        // The backup is valid! Restore it!
        Command_RestoreBackup();
      }
  }
```

C Copy Down

- Is the C copy down being performed?

```
void __copy_rom_sections_to_ram(void)
{

    int             index;

    if (__S_romp == 0L) return;

    /*
     *  Go through the entire table, copying sections from ROM to RAM.
     */
    for (index = 0;
         __S_romp[index].Source != 0 ||
         __S_romp[index].Target != 0 ||
         __S_romp[index].Size != 0;
         ++index)
    {
        __copy_rom_section( __S_romp[index].Target,
                            __S_romp[index].Source,
                            __S_romp[index].Size );

    }
}
```

Write Once Registers

- Are any write-once registers trying to be written by both the boot-loader and the application?
  - Watchdog Timers
  - Processor Mode registers
  - Memory registers

Example Test Cases

Test Case # : 003		
Objective: Verify that the s-record file can be loaded into the boot-loader GUI.		
Precondition: - Boot-loader Loaded - Identify command sent	**Input:** - Click Load File	**Expected Results:** - S-Record is loaded
Actual Result: S-Record is successfully loaded.		
Tester: **Jacob Beningo**	**Date:** **05/12/2014**	

Example Test Cases

Test Case # : 006		
Objective: Verify that the application checksum is written to flash.		
Precondition: - Boot-loader Loaded - Identify command sent - S-Record Loaded - Flash Erased - Application Written	**Input:** - Click Checksum Button	**Expected Results:** - Checksum is written to flash
Actual Result: Verified using the memory tool that the application checksum is calculated and written to flash.		
Tester: Jacob Beningo	**Date:** 05/12/2014	

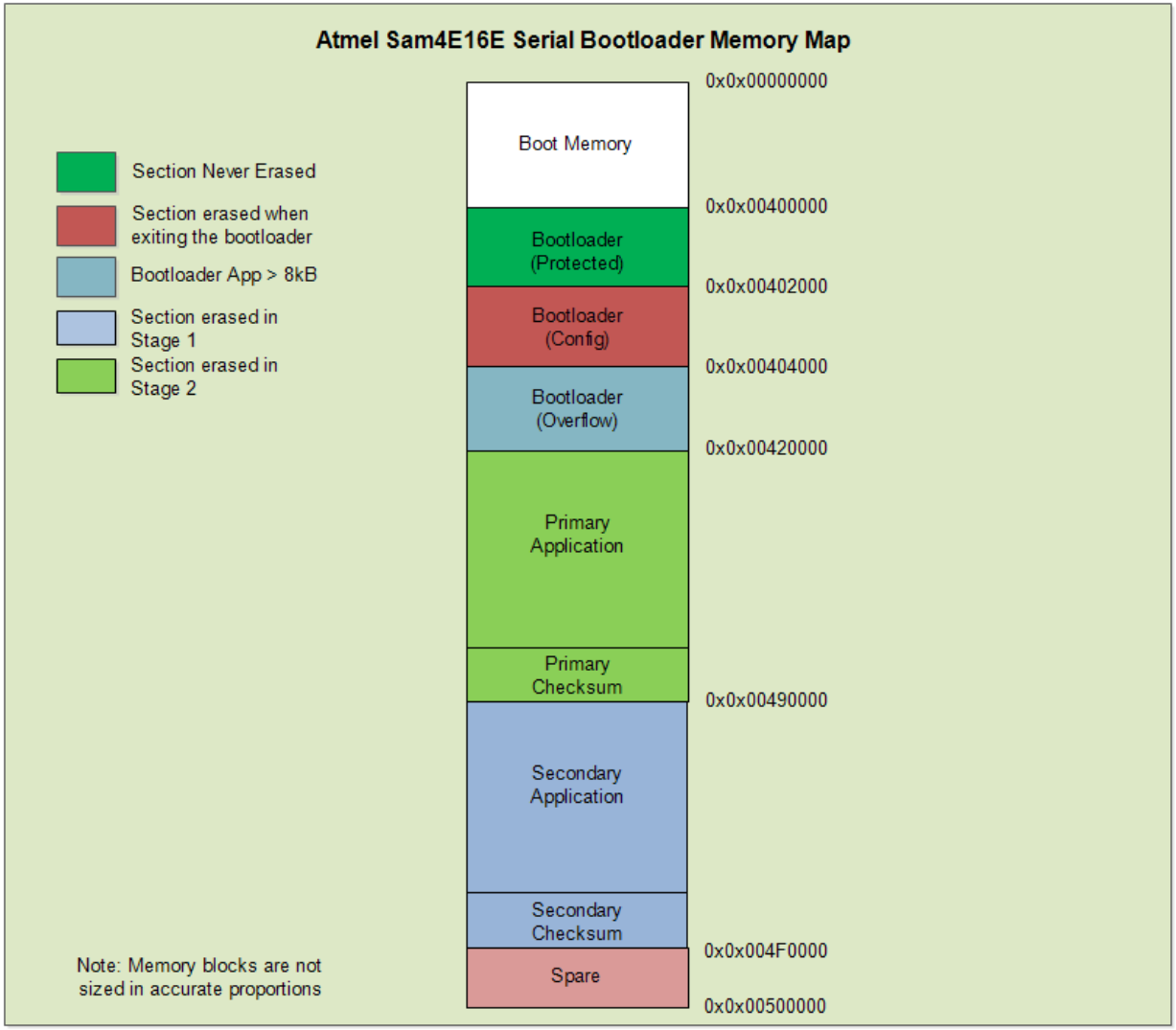
Example Test Cases

Test Case # : 009		
Objective: Verify that when flashing the unit if checksum is not calculated that the application does not load.		
Precondition: - Boot-loader Loaded - Application Loaded	**Input:** - Exit before checksum	**Expected Results:** - Boot-loader runs because the application checksum fails
Actual Result: Verified using led's that the boot-loader runs at start-up due to invalid checksum.		
Tester: **Jacob Beningo**	**Date:** **05/12/2014**	

Best Practices

- Test the corner cases
- Use a test harness
- Don’t be shy when it comes to error codes
- Use assertions
- Start with a simple test application
- Leverage example code but build in robustness

Bootloader Fall Backs



Course Review

- Bootloader Models and Concepts
- Interface Protocol Design
- Setting Up a Test Application
- Bootloader Implementation
- Troubleshooting Techniques

Where to go from here?

- Encryption
- Authentication
- Relocatable applications
- GUI Investigations
- Write your own

Additional Resources

- Download Course Material for
  - Updated C Doxygen Templates (Sept 2015)
  - Example source code
  - Bootloader White Paper
  - Templates
- Microcontroller API Standard
- EDN Embedded Basics Articles
- Embedded Bytes Newsletter

From www.beningo.com under

- Blog and Articles > Software Techniques > CEC Bootloader Design for MCUs

Jacob Beningo
Principal Consultant

P.O. Box 400
Linden, Michigan 48451

www.beningo.com

E : jacob@beningo.com
T : 810-844-1522
: Jacob_Beningo
: Beningo Engineering
: JacobBeningo
EDN : Embedded Basics

IEEE computer society Certified Software Development Professional

ARM Connected Community

Newsletters

- Embedded Bytes

http://bit.ly/1BAHYXm

Training

- MicroPython
- Bootloaders
- Low Power Design
- Resume Workshop
- Embedded Techniques

Presented by:

DesignNews

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS