Application Report
SPNA232A – July 2016 – Revised October 2019

Hercules™ AJSM Unlock

ABSTRACT

This application note explains how to utilize the features of the Advanced JTAG Security Module (AJSM) contained on Hercules devices. It is the responsibility of the developer to protect the AJSM key. Failure analysis is not an option, unless the AJSM key is provided. The lock can be changed, but that information should be provided to TI, if possible.

Project collateral and source code mentioned in this document can be downloaded from the following URL: http://www.ti.com/lit/zip/spna232.

Contents

List of Figures

Trademarks

Hercules, Code Composer Studio are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

1 AJSM Features

The AJSM has four features: a TIprogrammed visible unlock code, a customer defined 128-bit key protected by ECC, a scan path for unlocking the JTAG scan chain, and a key for permanently locking a device. The devices shipped from TI are programmed with the visible unlock code; any other value will result in locking a device. You can lock the device by programming any number of bits and your ECC. If all the bits are programmed (0), the device will be permanently locked.

1.1 AJSM Key Generation

The AJSM key is stored in memory that only allows changing 1s to 0s. TI ships the devices programmed with the 128-bit visible unlock code at address 0xF00000000 and your ECC.

```
TI Visible Unlock Code (TMS570 [Big Endian] ECC) 0xF00000000: 0xEFFDFFFF 0xFFFFFFFF 0xFFFDFFFE
0xFFEFFFFF ECC: 0xFFFF
```

This code is mainly 1s to allow for a custom key effective to 123 bits. The ECC is calculated based on the device's endianness. Hercules TMS570 devices are big endian, their visible unlock code ECC has been designed to be all 1s so any key can be selected. Hercules RM devices are little endian, their visible unlock code ECC will be 0xEDCO. Because the ECC is not all 1s, some custom unlock code values will not be possible. Read the value at 0xF000000 for your devices specific AJSM unlock code and 0xF0040000 for the ECC.

1.1.1 RM [Little Endian] Devices

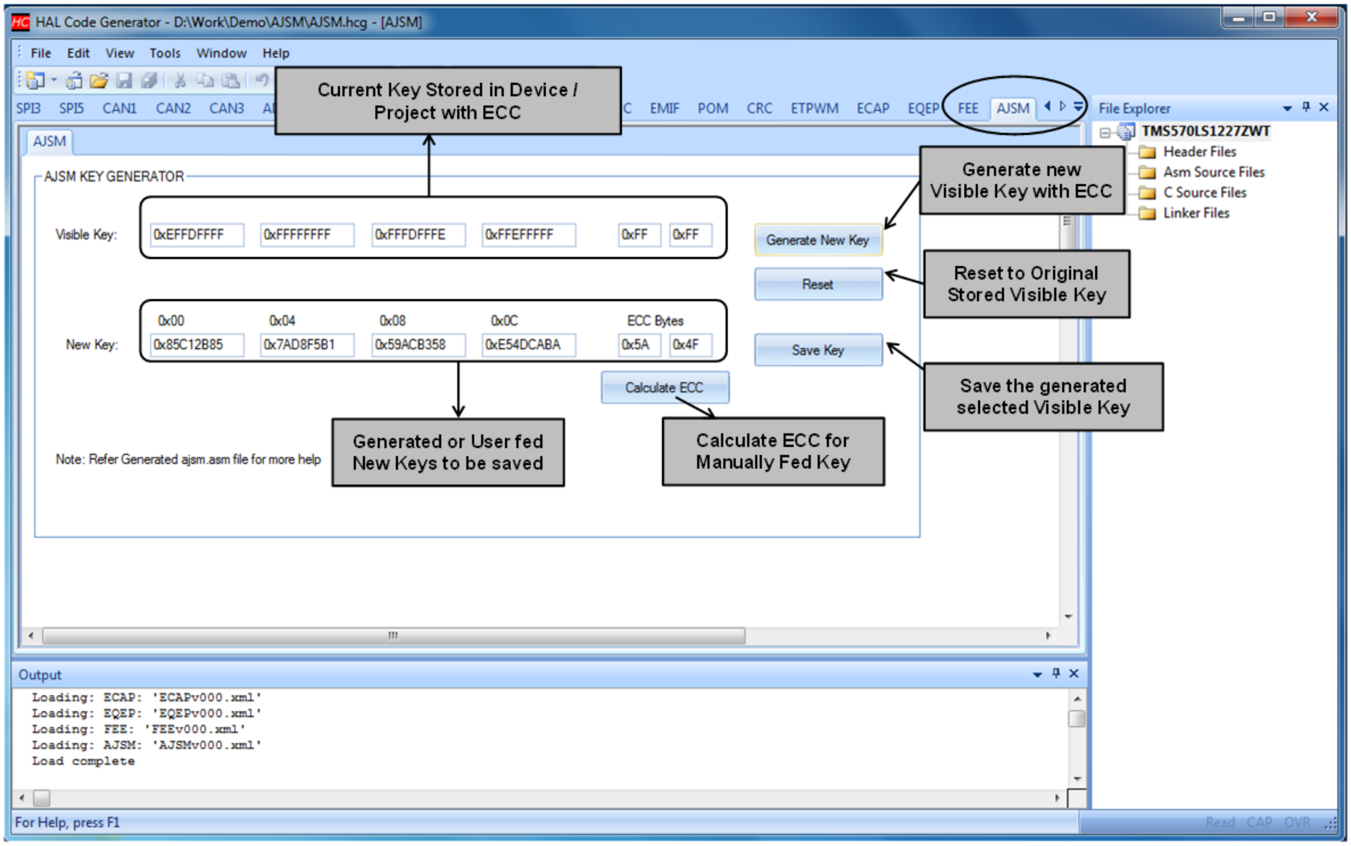
Because the ECC is calculated for the device endianness, the ECC for little endian devices will be 0xEDC0, limiting the custom key effective to 115 bits.

1.1.2 RM57 and TMS570LC4x Devices

There is an erratum for silicon versions before Rev. B that limits the custom key effective to 54 bits. Rev. B silicon was fixed so that the visible unlock code was the same as the other devices in the family and the ECC is 0xEDED for both big and little endian devices. The effective number of bits for Rev. B and later RM57 and TMS570LC4x devices is 119 bits.

1.1.3 HALCoGen Key Generation

HALCoGen supports AJSM Key generation from HALCoGen version 4.06.00.



Current Key Stored in Device / Project with ECC

Generate new Visible Key with ECC

Reset to Original Stored Visible Key

Save the generated selected Visible Key

Generated or User fed New Keys to be saved

Calculate ECC for Manually Fed Key

	0x00	0x04	0x08	0x0C	ECC Bytes	
Visible Key:	0xEFFDFFFF	0xFFFFFFFF	0xFFFDFFFE	0xFFEFFFFF	0xFF	0xFF
New Key:	0x85C12B85	0x7AD8F5B1	0x59ACB358	0xE54DCABA	0x5A	0x4F

Note: Refer Generated ajsm.asm file for more help

```
Loading: ECAP: 'ECAPv000.xml'
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
```

Figure 1. HAL Code Generator

The following steps have to be followed to generate AJSM Keys:

1. Create a New or Open existing HALCoGen Project with HALCoGen Version 4.06.00 or Greater.
2. Go to the “Driver Enable” tab and check 'Enable AJSM'.
3. Options:
   a. Option 1 – HALCoGen Generate Key AJSM Unlock Key:
      i. Select Generate New Key, notice “New Key” fields Including the ECC bytes get updated with randomly generated keys.
      ii. Once satisfied with the Key, choose “Save Key”. Notice the “New Key” gets saved to the “Visible Key”.
   b. Option 2 – Manually Feed Key:
      i. Feed in the necessary Keys directly to the “New Key” Fields.
      ii. Select Calculate ECC.
      iii. Once satisfied with the Key, choose “Save Key”. If the selected “New Key” does not flip “0” to “1“ with the current “Visible Key”, the “New Key” will replace the current “Visible Key”, otherwise you will be notified with an error message in the output window of HALCoGen. You might have to redo the Key selection or use “Generate New Key” to allow HALCoGen to provide a suitable key.
4. Generate the code. File “ajsm.asm” gets generated, which contains the necessary memory or data section with the new AJSM keys. This new ajsm keyis stored in HALCoGen project and the next time this same project is opened this key will appears in the “Visible Key” Field.
5. Memory section must be added to the linker file to program the "ajsm" data section to address location 0xF0000000 and “ajsmecc” data section to address location 0xF0400000, before building the project.

1.1.4 Care About's

- Necessary precautions must be taken to save this project file securely. The HALCoGen project file (*.dil) contains the new Key information in plain txt.
- If you have already programmed the Keys without using HALCoGen, the HALCoGen project will still have the default key. It is the user's responsibility to manually feed the current Keys using the “New Key” fields and Save it to reflect in the “Visible Key”.

1.2 Locking

The F021 Flash API supports programming the customer OTP of which the first 128 bits and their associated ECC are the AJSM Visible Unlock Code at address 0xF0000000. TI and third party tools support programming this location either integrated into the application or standalone. Standalone programming of the AJSM is recommended for production flows that need to perform system level testing.

Once a device is locked, the ICEPICK and AJSM will be the only visible tap (Test Access Port); the scan chain will still be available and discoverable as a length of 6 to allow for easy identification that the device is functional. When the AJSM is locked, the typical emulator response will be no communication. To differentiate JTAG communication issue from a locked AJSM, perform a scan chain check or read the JTAG ID.

1.3 Code Composer Studio™ (CCS) Temporarily Unlocking

The Hercules AJSM can be temporarily unlocked by scanning the 128-bit XOR of the customer code and the TI Visible Unlock Code in reverse 32-bit order (without ECC) on the AJSM tap. The device will remain unlocked through a system reset, but not power on reset.

```
Scan the 128 bit XOR of the TI Visible Unlock Code and the customer code in 32-
bit reverse order // 0xF0000000: 0xAECD0000AECD0001AECD0002AECD0003, // in reverse 32bit order
(without ECC) = 5122fffc5130fffc5132fffe4130ffff
```

1.3.1 Command Line

Dbgauth facilitates debugger authentication on supported Texas Instruments SoC platforms. You will provide options that define the SoC being authenticated (-c, -m, -t), options that define the debugger environment (-x), and options that define inputs into the authentication algorithm (-k).

Based on the supplied SoC type, options are classified as required, optional, or N/A.

dbgauth is available as part of CCS EMUPACK releases. It can be found in:

```
CCS - <ccs_install>\ccs_base\common\uscif
```

Example:

```
dbgauth -c testBoard.dat -s ajsm -t cortexr4 -k 00112233445566778899AABBCCDDEEFF -m 1
```

Use the -h option for details on how to use the dbgauth tool.

The board.dat file needs to include the AJSM which has a 4-bit IR and is available at ICEPick port address 18.

Open the xml target configuration file from the download link in the abstract with CCS and click the test connection button. A test connection window will pop up and list the path to the generated board.dat file. The default path will be C:\users\<account name>\AppData\Local\TEXAS~1\CCS\ti\<#>\<#>\BrdDat\testboard.dat. Every time the button is pressed, this file will be overwritten; save a copy, if desired.

1.3.2 GEL

The dbgauth tool can be integrated into CCS using GEL as a means of automatically unlocking the device as part of the startup. The following GEL source is an example of such an implementation:

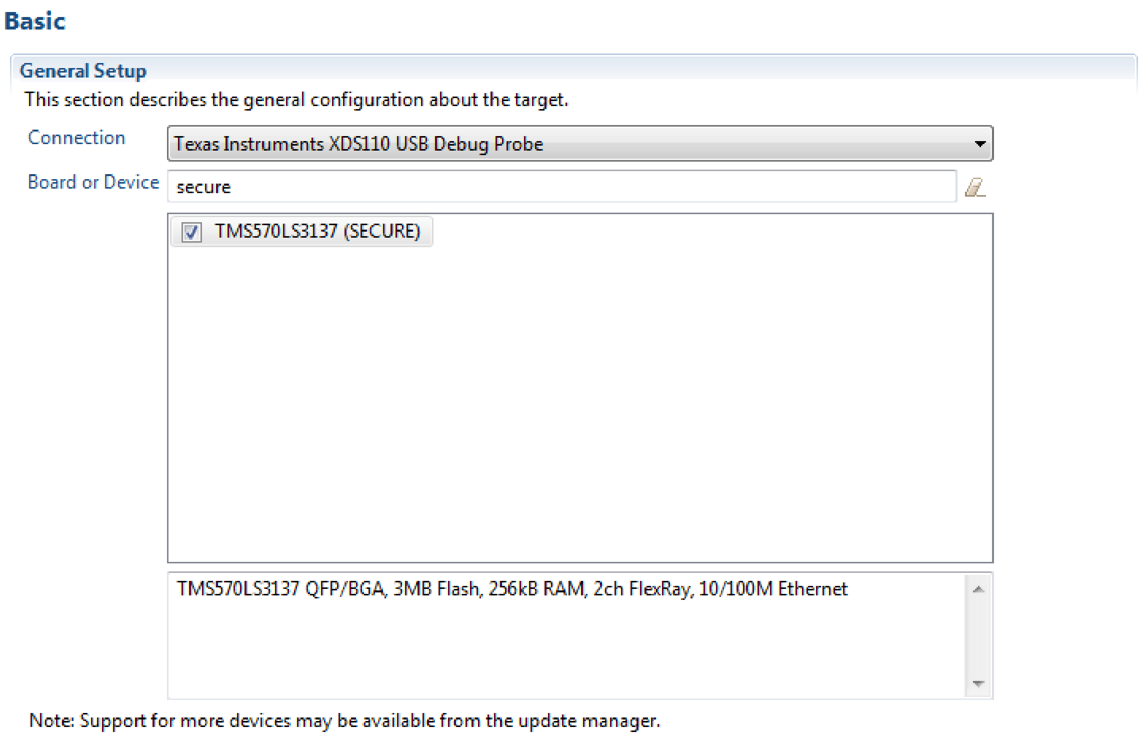
```
/*----------------------------------------------------------------------------*/
/* Function - StartUp() */
/* */
StartUp(){
AJSM_Unlock_Demo();
}

/* StartUp() */
/*----------------------------------------------------------------------------*/
/* MenuItem - "TMS570LS0714 AJSM Unlock Demo" */
/* */
menuitem "TMS570LS0714 AJSM Unlock Demo";
hotmenu AJSM_Unlock_Demo(){
//Customer OTP 0xF0000000: 0xAECD0000AECD0001AECD0002AECD0003
//Cust OTP ECC 0xF0040000: 0x6E71 - BE
// Scan the 128 Bit XOR of the TI Visible Unlock code
// 0xEFFDFFFFFFFFFFFFFFFDFFFEFFEFFFFF,
// in reverse 32bit order (without ECC) = 5122fffc5130fffc5132fffe4130ffff
GEL_System("C:\\ti\\ccsv6\\ccs_base\\common\\uscif\\dbgauth -
c C:\\Users\\XXXXXXXX\\AppData\\Local\\TEXASI~1\\CCS\\ti\\2\\0\\BrdDat\\ccBoard0.dat -s ajsm -
t cortexr4 -k 5122fffc5130fffc5132fffe4130ffff -m 1");
}
```

1.3.3 GUI Target Configuration Files

Code Composer Studio version 6.1 and greater with TI emulators version 6.0.394.0 and greater have support for AJSM unlock. In the attached zip file, there are new versions of devices with “_secure” that can be selected when creating a target configuration. For example TMS570LS3137_secure.

The new secure target configurations will have the AJSM module included in the scan chain. The AJSM module has a field for the AJSM key and will scan this value after ICEPICK connect before connecting to the Cortex. This unlock is temporary and will be re-evaluated every time the AJSM module is refreshed either by a reset or by scanning values into the AJSM Key.

Extract the files in spna232.zip to the device folder of Code Composer Studio. The standard path is C:\ti\ccsv6\ccs_base\common\targetdb\devices\. If the desired part number does not exist, follow the instructions in the readme.txt file to generate it.



Basic

General Setup

This section describes the general configuration about the target.

Connection: Texas Instruments XDS110 USB Debug Probe

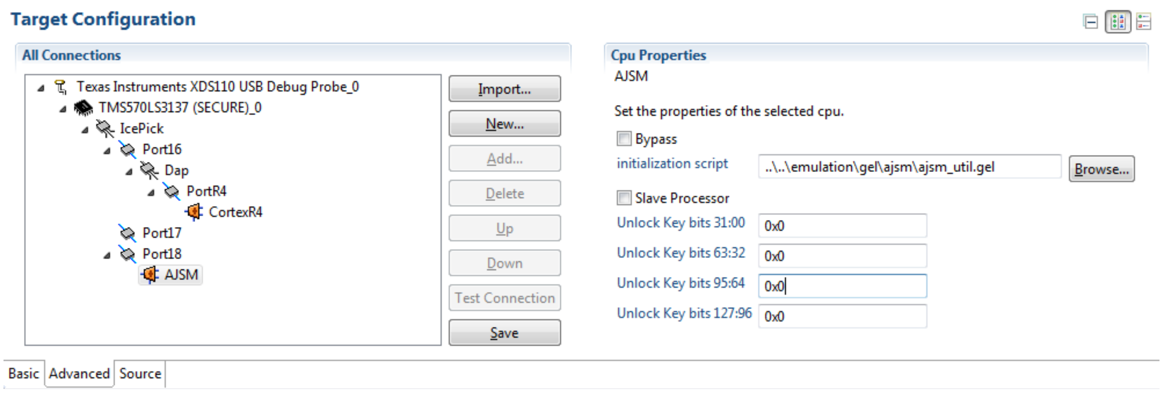
Board or Device: secure

☑ TMS570LS3137 (SECURE)

TMS570LS3137 QFP/BGA, 3MB Flash, 256kB RAM, 2ch FlexRay, 10/100M Ethernet

Note: Support for more devices may be available from the update manager.

Figure 2. Target Configuration – Basic



Target Configuration

All Connections

```
Texas Instruments XDS110 USB Debug Probe_0
  TMS570LS3137 (SECURE)_0
    IcePick
      Port16
        Dap
          PortR4
            CortexR4
      Port17
      Port18
        AJSM
```

Cpu Properties

AJSM

Set the properties of the selected cpu.

□ Bypass

initialization script: ..\..\emulation\gel\ajsm\ajsm_util.gel

□ Slave Processor

Unlock Key bits 31:00: 0x0

Unlock Key bits 63:32: 0x0

Unlock Key bits 95:64: 0x0

Unlock Key bits 127:96: 0x0

Figure 3. Target Configuration – Advanced

2 References

- *JTAG Programmer Overview for Hercules-Based Microcontrollers* (SPNA230)

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (July 2016) to A Revision	Page
• Added new information to Section 1.1.3	2

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265