

Customer Notification

RH850TM Series

Y-GHS-MULTI-V800

Operating Precautions

GHS MULTI 2000 Integrated Development Environment Version 6.X, 7.X

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

"Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority- owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

Table of Contents

- (A) Table of Operating Precautions5
- (B) Installing Patches under Multi V6.1.x or later.....7
- (C) Description of Operating Precautions8
- (D) Valid Specification28
- (E) Revision History30

(A) Table of Operating Precautions

Anchor

No.	Outline						
		Version	6.1.4/ 2013.5.5	6.1.6/ 2015.1.7	7.1.4/ 2017.1.5	7.1.6/ 2018.1.5	7.1.6/ 2019.1.5
a106	Linker does not delete unused functions	x	✓	✓	✓	✓	✓
a107	Problem with hex constants in ease850	x	✓	✓	✓	✓	✓
a108	Compiler interrupt routines do not save FPSR and FPEPC	x	✓	✓	✓	✓	✓
a109	Linker invokes unknown section SHN_ABS	x	✓	✓	✓	✓	✓
a110	Multi IDE crash in New Project Wizard with MultiCoreArchive projects	x	✓	✓	✓	✓	✓
a111	Invalid bitfield handling in optimized If/else clause	✓	✓	✓	✓	✓	✓
a112	P26: TOOLS-20948 write to global variable was wrongly moved before __DI	x	✓	✓	✓	✓	✓
a113	P23: BTO-2134 fix error in code motion optimization	x	✓	✓	✓	✓	✓
a114	P26: BTO-2144 update global loop variables in nested loops	x	✓	✓	✓	✓	✓
a115	P25: BTO-2136/ .ghsexports with library; BTO-2139 exportall should not imply extractall	x	✓	✓	✓	✓	✓
a116	P22: BTO-2095 ease850 now recognizes bt and bf	x	✓	✓	✓	✓	✓
a117	P21: BTO-2089 accurate simulation of ldsr FPST and ldsr FPSFG	x	✓	✓	✓	✓	✓
a118	P36: BTO-2619 improves the linker	x	✓	✓	✓	✓	✓
a119	850eserv2 Debug Server Update v2.037	x	x	✓	✓	✓	✓
a120	P39: Compiler update generates syncp	x	✓	✓	✓	✓	✓

Operating Precautions for Y-GHS-MULTI-V800™

No.	Outline	Version					
		6.1.4/ 2013.5.5	6.1.6/ 2015.1.7	7.1.4/ 2017.1.5	7.1.6/ 2018.1.5	7.1.6/ 2019.1.5	7.1.6/ 2020.1.5
a121	BTO-2844: Invalid offset for member of struct with constant address	x	x	✓	✓	✓	✓
a122	Introducing new V2015.1.7 compiler	-	x	✓	✓	✓	✓
a123	TOOLSCL-604 assignment to multiple bitfields fails	x	x	✓	✓	✓	✓
a124	TOOLSCL-628 Local copy of function pointer overwritten, then reused	✓	x	✓	✓	✓	✓
a125	Option -section bss= with C++ not working	x	✓	✓	✓	✓	✓
a126	Division by zero was moved outside	x	x	✓	✓	✓	✓
a127	Introduction of Multi V7.1.4/2017.1.5	-	-	x	✓	✓	✓
a128	Unreached assembly routines deleted by linker	-	-	x	✓	✓	✓
a129	Dongle License issue	-	x	x	x	x	x
a130	Pointer dereference illegally pulled out of 'if'	-	x	x	✓	✓	✓
a131	License Manager Improvement	-	x	x	✓	✓	✓
a132	Introduction of Multi V7.1.6/2020.1.5	-	-	-	-	-	x

- ✓ Not applicable
- x Applicable
- Not checked

Note: The rank is indicated by the letter appearing at the 5th position from the left in the lot number, marked on each product.

(B) Installing Patches under Multi V6.1.x or later

Occasionally, you may need a patch to update a Green Hills Product. Patches are installed with a utility called gpatch and have a filename indicating the patch number and possibly the host architecture and product the patch is for. As an example, a theoretical patch_12345_win32_comp.iff would be patch number 12345 for the Green Hills Compiler hosted on Microsoft Windows.

Patches are generally provided by Green Hills Technical Support to address a specific issue or to add support for a custom feature (e.g., to add an INTEGRITY BSP that isn't part of the standard product).

Patches are not typically used to upgrade from one version of a product to another. To upgrade, use the installer for the new product version.

New releases of Green Hills Products may include patches to update other products to work with them. For example, a new INTEGRITY release may include a patch for MULTI or the Compiler. Required patches will be listed in the release notes.

When installing patches, you should always use the most recent version of the gpatch utility, which is always available from the downloads section of support.ghs.com, or via FTP at http://ftp.ghs.com/private/patch_installers/. Download the file appropriate for your host system, and copy the gpatch executable to your MULTI or Compiler directory. It's a good idea to back up the original.

Installing MULTI / Compiler Patches

Patches for MULTI or the Compiler should be installed into the root of the installation directory. For example, if MULTI 6.1 is installed at c:\ghs\multi_61\, and you receive a patch for this version of MULTI, first copy the patch file to C:\ghs\multi_61\patch_#####.iff (substituting the ##### for the actual patch number). Next, cd to the MULTI directory and run gpatch:

```
c:\> cd c:\ghs\multi_61\
```

```
c:\ghs\multi_61> gpatch.exe patch_#####.iff
```

The MULTI 6.1 and Compiler 2012.1 releases (and later) are installed into separate directories. It's important that patches are installed in the proper directory. The support or sales engineer who sends you the patch will let you know which directory to install into.

(C) Description of Operating Precautions

<p>No. a106</p>	<p><i>Linker does not delete all unused functions</i></p>
	<p><u>Version Information</u> V2013.5.5</p> <p><u>Details:</u> Along with the introduction of the assembler switch statement in rh850 cores, the linker was not able to recognize unused references.</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> ccv850 -cpu=rh850 -O -o test.out -delete *.o</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch Y-GHS-MULTI-V800-FULL-V614_2013.5.5-PATCH02</p>

No. a107	Problem with hex constants in ease850
	<p><u>Version Information</u> V2013.5.5</p> <p><u>Details:</u> Using -new_assembler, certain integer constants give 'out of range' or 'expected a register' parsing errors for the sub and add instructions.</p> <pre> sub 0xffffffff,r7 sub 0xffffedc8,r7 [ease850] (error) file_mainASM.850 868: expected a register add - (0xffffffff) , r7 -----^ [ease850] (error) file_mainASM.850 924: out of range addi - (0xffffedc8) , r7 , r7 -----^ [ease850] (error) errors during processing </pre> <p><u>Example:</u> None.</p> <p><u>Command Line</u> ccv850 -cpu=rh850 -O -o test.out -delete *.o</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch.</p>

No. a108	Compiler interrupt routines do not save FPSR and FPEPC
	<p><u>Version Information</u> V6.1.4 all versions</p> <p><u>Details:</u> P10: BTO-1906: Interrupt services implemented with the compiler do not save FPU related status register FPSR and FPEPC.</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> ccv850 -cpu=rh850 -O -o test.out</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch Y-GHS-MULTI-V800-FULL-V614_2013.5.5-PATCH02</p>
No. a109	Linker invokes unknown section SHN_ABS
	<p><u>Version Information</u> V6.1.4 all versions</p> <p><u>Details:</u> When inline assembler of the compiler is using a jump relative instruction, such as</p> <pre>asm (" jr _START_TIMERM") ;</pre> <p>The linker "elxr.exe" claims for unknown section SHN_ABS.</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> ccv850 -cpu=rh850 -O -o test.out</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch Y-GHS-MULTI-V800-FULL-V614_2013.5.5-PATCH02</p>

No. a110	Multi IDE crash in New Project Wizard with MultiCoreArchive projects
	<p><u>Version Information</u> V6.1.4 all versions</p> <p><u>Details:</u> Sometimes, the NPW crashes, if a new object is inserted in the Multi project manager.</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> Multi.exe</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch Y-GHS-MULTI-V800-FULL-V614_2013.5.5-PATCH02</p>

No. a111	Invalid bitfield handling in optimized if/else clause
	<p><u>Version Information</u> V5.1.7D, V2012.x, V2013.1.x</p> <p><u>Details:</u> The broken optimization combines assignments to bitfields so that a single load-modify-store sequence can perform multiple assignments. The issue involves an incorrect interpretation of control flow.</p> <p><u>Example:</u></p> <pre>extern int printf(const char *, ...); struct { unsigned int b1:1; unsigned int b2:1; unsigned int b3:1; unsigned int b4:1; unsigned int b5:1; } s = { 0 }; unsigned char flag=0, count=0 ; void func(void) { if (flag == 3) { s.b4 = 0; if (count == 1) flag = 2; } else s.b1 = 0; s.b5 = 1; } int main(void) { func(); if (s.b5 != 1) printf("%s: b5 is false\n", __FILE__); return 0; }</pre> <p><u>Command Line</u> ccv850 -cpu=v850e2 -G -Ospace test.c</p> <p><u>Workaround</u> For V6.1.4/2012.5.5, please use latest patch Y-GHS-MULTI-V800-FULL-V614_2012.5.5-PATCH01</p> <p>For V6.1.4/2013.1.5, please use latest patch Y-GHS-MULTI-V800-FULL-V614_2013.1.5-PATCH02</p> <p>In case of V5.x.x: The optimization limitation can be workaround by using special compiler option -Z3245 or -Z990 In general, the limitation can be removed, if the variable is implemented/declared with 'volatile' attribute.</p>

No. a112	P26: TOOLS-20948 write to global variable was wrongly moved before __DI
<p><u>Version Information</u> V6.1.4/2013.5.5</p> <p><u>Details:</u> TOOLS-20948 write to global variable was wrongly moved before __DI The problem here is that the write to variable “Glob” between the DI/EI pair is moved outside the DI/EI by the codemotion optimization.</p> <p><u>Example:</u></p> <pre>#include <stdio.h> volatile int Flag=1; volatile int Glob=0; int a = 1; int main() { if (Flag==1) { __DI(); Glob=a; __EI(); } else Glob=a; printf("Hello world. %d \n", Glob); return 0; }</pre> <p><u>Command Line</u> ccrh850 -cpu=rh860 -c -O</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch Y-GHS-MULTI-V800-FULL-V614_2013.5.5-PATCH02 plus v800-v2013.5.5-comp-P26 or later</p>	

No. a113	P23: BTO-2134 fix error in code motion optimization
	<p><u>Version Information</u> V6.1.4/2013.5.5</p> <p><u>Details:</u> BTO-2134 fix error in code motion optimization An optimization which moves code outside of a conditional, such as “if-then-else”, is performed in cases where it is not entirely safe, resulting in incorrect behaviour in very rare cases. Below is reduced from the example originally reported by the customer.</p> <p><u>Example:</u></p> <pre>void test(int flag, unsigned char start) { unsigned char u; for (u=start; u<15; u++) if (flag) Display(u,u+1); else Other(u,u+1); }</pre> <p><u>Command Line</u> ccrh850 -cpu=rh860 -c -O</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch v800-v2013.5.5-comp-P26 or later</p>

No. a114	P26: BTO-2144 update global loop variables in nested loops
	<p><u>Version Information</u> V6.1.4/2013.5.5</p> <p><u>Details:</u> BTO-2144 update global loop variables in nested loops; An optimization for simple, nested loops did not check for the case where one or both of the loop control variables are defined outside of the function and the inner loop contains a function call which might use the variables. The global loop control variables were not updated within the loop and therefore the values were wrong when the function was called.</p> <p><u>Example:</u></p> <pre>unsigned char i, j; unsigned char array[32][32]; void func(void) { array[origi][origj] ++; } void test(void) { for (i = 0; i < 8; i ++) for (j = 0; j < 16; j ++) func(); }</pre> <p>With simple, nested loops something very close to the code above is meant.</p> <p><u>Command Line</u> ccrh850 -cpu=rh860 -c -O</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch v800-v2013.5.5-comp-P26 or later</p>

No. a115	P25: BTO-2136/ .ghsexports with library; BTO-2139 exportall should not imply extractall
	<p><u>Version Information</u> V6.1.4 all versions</p> <p><u>Details:</u> P25 gcores update: BTO-2136: now works properly along with “.ghsexports”- file using a library; BTO-2139: the exportall option does not imply extractall</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> gcores core1 core2 share -o allcores</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use latest patch v800-v2013.5.5-comp-P25</p>

No. a116	P22: BTO-2095 ease850 now recognizes bt and bf
	<p><u>Version Information</u> V6.1.4 all versions</p> <p><u>Details:</u> P22 assembler update: BTO-2095, the assembler ease850 now recognizes bt and bf instructions.</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> ease850 test.850</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use patch v800-v2013.5.5-comp-P22</p>

<p>No. a117</p>	<p>P21: BTO-2089 accurate simulation of “ldsr FPST” and “ldsr FPSFG”</p>
	<p><u>Version Information</u> V6.1.4 all versions</p> <p><u>Details:</u> P21 simulator update simrh850 contains multiple issues: BTO-1935 corrected simulation of expression “0x80000000 % -1”. BTO-2005 simrh850 ignored software breakpoint when no .syscall section in program. BTO-2089 accurate simulation of “ldsr FPST” and “ldsr FPSFG”.</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> MULTI> connect simrh860 –cpu=rh850g3m test.out</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use patch v800-v2013.5.5-P21</p>
<p>No. a118</p>	<p>P36: BTO-2619 improves the linker</p>
	<p><u>Version Information</u> V6.1.4/2013.5.5</p> <p><u>Details:</u> BTO-2619 improves the linker optimization of converting 23-bit SDA access to 16- bit addressing mode wherever possible.</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> ccrh850 –cpu=rh850 –o test test.c –large_sda –shorten_loads</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use patch v800-v2013.5.5p-comp-P36</p>

No. a119	850eserv2 Debug Server Update v2.037
	<p><u>Version Information</u> V6.1.4/2013.5.5 and later</p> <p><u>Details:</u> Improves debug capabilities of the Multi debugger with various RH850 targets using Renesas emulators.</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> None.</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use patch v800-v2015.1.3p-comp-P06 Only to be used with latest Exec RH850_EXEC_E2.31.00.02</p>

No. a120	P39: Compiler update generates syncp
	<p><u>Version Information</u> V6.1.4/2013.5.5</p> <p><u>Details:</u> A new option "-rh850_eint_erratum" is provided. This option has no effect if the core is not one of the RH850 cores. This option is on by default if the core is RH850G3M and can be disabled with "-no_rh850_eint_erratum".</p> <p>With "-rh850_eint_erratum", for the following #pragma intvect directive,</p> <pre>#pragma intvect function integer_constant</pre> <p>"syncp" instruction is generated immediately the jump to an interrupt handler:</p> <pre>syncp jr _function</pre> <p><u>Example:</u> None.</p> <p><u>Command Line</u> None.</p> <p><u>Workaround</u> For V6.1.4/2013.5.5, please use patch v800-v2013.5.5p-comp-P39</p>

No. a121	BTO-2844: Invalid offset for member of struct with constant address
	<p><u>Version Information</u> V6.1.4/2013.5.5, V6.1.6/2015.1.7</p> <p><u>Details:</u> The compiler does not calculate correct member offsets in structures allocated by the '#pragma ghs io' directive.</p> <p>Example: <pre>typedef struct { long fld1[250]; long fld2[10]; } DMASS_T;</pre> <pre>#pragma ghs io DMASS 4000 extern DMASS_T DMASS;</pre> <pre>long * A(int ch) { return DMASS.fld2 + ch; } // A() must be before f() in order to fail long * f(int ch) { return DMASS.fld1 + ch; }</pre> </p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> None.</p> <p><u>Workaround</u> Please use pointers to the allocated structures, instead of the structure itself.</p> <p>For V6.1.4/2013.5.5, please use patch v800-v2013.5.5p-comp-P40</p> <p>For V6.1.6/2015.1.7, please use patch v800-v2015.1.7p-comp-P01</p>

No. a122	<i>Introducing new V2015.1.7 compiler</i>
	<p><u>Version Information</u> V6.1.6/20151.7</p> <p><u>Details:</u> The compiler V2015.1.7 is released and available for download for registered users of the Renesas Toolweb. A new license key might be required and is available from Green Hills (eurolicense@ghs.com).</p> <p><u>Example:</u> None.</p> <p><u>Command Line</u> None.</p> <p><u>Workaround</u></p>

No. a123	TOOLSCL-604 assignment to multiple bitfields fails
	<p><u>Version Information</u></p> <p>V5.3.0, V6.1.4/2012.5.5, V6.1.42013.5.5, V6.1.6/2015.1.7</p> <p><u>Details:</u></p> <p>Assigning the value of two or more bitfields in the same struct, where each bitfield has a single bit, can fail in two cases.</p> <p>Case 1) if the values are all taken from bits in a single variable or struct and at least one of the values is negated during the assignment.</p> <p>Case 2) The constant value 0 is stored to one field and a single bit is stored to another field, whether or not the single bit is negated.</p> <p>Case 2 only happens with V850E3V5 or RH850 and only in v2014.1 and later.</p> <p>Forms of negation that are involved in the bug include</p> <pre>x.A = !y.A; x.B = (y.B == 0); x.C = (y.C != 1); x.D = (y.D == 0) ? 1: 0; x.E = (y.E != 1) ? 1: 0;</pre> <p>Optimization levels -Osize -Ogeneral -Ospeed are affected, but not -Odebug or -Onone.</p> <p><u>Example:</u></p> <pre>#include <stdio.h> struct S { unsigned A:1, B:1, C:1; } x = {0, 0, 0}, y = {0, 0, 1}; void case1(struct S *p1, struct S *p2) { p1->A = !p2->A; p1->B = (p2->B == 0); /* equivalent to ! in this case */ } void case2(struct S *p1, struct S *p2) { p1->A = p2->A; p1->B = p2->B; p1->C = 0; } int main() { case1(&x, &y); if (x.A != 1 x.B != 1) printf("x = {%d, %d}, not {1, 1}\n", x.A, x.B); case2(&x, &y); if (x.A != 0 x.B != 0 x.C != 0) printf("x = {%d, %d, %d}, not {0, 0, 0}\n", x.A, x.B, x.C); return 0; }</pre>

No. a123	<i>TOOLSCL-604 assignment to multiple bitfields fails (continued ...)</i>
	<p><u>Command Line</u> Compile ccrh850 -O test.c Run Debugger with simrh850 a.out</p> <p><u>Workaround</u> Use the tilda ~ or ^ operators to negate the value instead of ! or comparison to 0.</p> <p>For V2013.5.5 please use the patch v800-v2013.5.5p-comp-P41 For V2015.1.7 please use the patch v800-v2015.1.7p-comp-P08</p>

No. a124	<i>TOOLSCL-628 Local copy of function pointer overwritten, then reused</i>
	<p><u>Version Information</u> V6.1.6/2015.1.7</p> <p><u>Details:</u> This is related to a function call made through an integer constant. The compiler did not keep track of the fact that the call to the function modified the register used to make the call. A later use of same integer constant would assume the integer was in the register, but it had been lost. The issue is raised with optimizations '-Osize', '-Ospeed', '-Ogeneral'.</p> <p><u>Example:</u></p> <pre>typedef int (* ptrfunc)(void); int gbl_flg = 0; void func1(void) { int tmp = ((ptrfunc)0x1234678U)(); gbl_flg = 1; while (tmp > 0) tmp = ((ptrfunc)0x1234678U)(); }</pre> <p><u>Command Line</u> Compile with any optimization '-Osize', '-Ospeed', '-Ogeneral'.</p> <p><u>Workaround</u> None.</p> <p>For V2015.1.7 please use the patch v800-v2015.1.7p-comp-P11</p>

No. a125	Option -section bss= with C++ not working
	<p><u>Version Information</u> V6.1.4/2013.5.5</p> <p><u>Details:</u> This issue is involving the option -bss= when programming in C++. The operation results in a fatal error, so that a user would know about any issue with this.</p> <p><u>Example:</u> None</p> <p><u>Command Line</u> Compile ccrh850 -O test.c -bss=".mybss"</p> <p><u>Workaround</u> None.</p> <p>For V2013.5.5 please use the patch v800-v2013.5.5p-comp-P42</p>
No. a126	Division by zero was moved outside
	<p><u>Version Information</u> V6.1.4/2013.5.5, V6.1.6/2015.1.7</p> <p><u>Details:</u> Division by zero was moved outside of an 'if' statement, so that the division might happen even when the 'if' condition is false. The generated code might perform the division even when x is 0.0, resulting in an exception on RH850 targets with H/W FPU support. The issue is raised only with optimization '-Osize', '-Ospace', '-OS'.</p> <p><u>Example:</u> <pre> if (x != 0.0) x = y / x; </pre> </p> <p><u>Command Line</u> Compile ccrh850 -Osize test.c</p> <p><u>Workaround</u> None.</p> <p>For V2013.5.5 please use the patch v800-v2013.5.5p-comp-P42 For V2015.1.7 please use the patch v800-v2015.1.7p-comp-P11</p>

No. a127	Introduction of Multi V7.1.4/2017.1.5
	<p><u>Version Information</u></p> <p>V7.1.4/2017.1.5</p> <p><u>Details:</u></p> <p>MULTI 7 is a major release</p> <ul style="list-style-type: none"> • Library improvements faster sin, cos, exp, log • Additional SDA base registers -additional_sda_reg=n with 5 additional SDA base registers • Improved RAM Initialization all initialized as 32-bit to avoid ECC errors on RH850 device family • EABI changes (since v2016.5) • -registermode=common (since v2016.5) • C++11 support (since v2016.5) • Argument Type Checking (since v2016.5) • Improved Download to RAM Dialogue in Multi Debugger <p><u>Example:</u></p> <p><u>Command Line</u></p> <p><u>Workaround</u></p>

No. a128	Unreached assembly routines deleted by linker
	<p><u>Version Information</u> V7.1.4/2017.1.5</p> <p><u>Details:</u> Code generated by the assembler might be deleted by the link process, if no reference to the entry is found. That means, that for example interrupt vectors are not properly inserted.</p> <p><u>Example:</u></p> <p><u>Command Line</u></p> <p><u>Workaround</u> Install latest GHS patch v800-2017.1.5</p>

No. a129	Dongle License issue
	<p><u>Version Information</u> V7.1.4/2017.1.5, V7.1.6/2018.1.5</p> <p><u>Details:</u> Fixes a problem where a safe-net hardware key (dongle) may not be recognized with Windows Spring Creators update (version 1803) or beyond. This should be applied to MULTI 7.1.6 and to compiler 2016.5.4-2018.1.4 releases.</p> <p><u>Example:</u></p> <p><u>Command Line</u></p> <p><u>Workaround</u> Install latest GHS patch from here</p>

No. a130	Pointer dereference illegally pulled out of 'if'
	<p><u>Version Information</u> V6.1.6/2015.1.7, V7.1.4/2017.1.5</p> <p><u>Details:</u> There must be nested if statements where the control expression of inner if statement involves a memory access and the inner if statement can be rearranged into an assignment such as $x = a ? b : c$.</p> <p>An optimization (-Ospeed or -Ogeneral) sometimes moves the evaluation of expressions from within the body of an if statement to before the if statement if it can be done safely and correctly. The optimization should not be performed if the control expression of the inner if statement accesses memory that would not otherwise be accessed unconditionally.</p> <p><u>Example:</u> Below example shows only the principle of the issue, any change in the code and compile options may not reflect the issue anymore.</p> <pre> unsigned int *p = 0; int main() { int retVal = 9; if(p) { if(*p > 0) /* Cause a NULL pointer dereference */ retVal = 3; else retVal = 6; } return retVal; } </pre> <p><u>Command Line</u> ccrh850 -cpu=... -Ospeed ... or ccrh850 -cpu=... -Ogeneral ...</p> <p><u>Workaround</u> Install latest GHS patch v800-2017.1.5-P10 and v800-v2015.1.7-P16</p>

No. a131	License Manager Improvement
	<p><u>Version Information</u> V7.1.4/2017.1.5, V7.1.6/2015.1.5</p> <p><u>Details:</u> IDE patch to license manager to accept either ip address or username.</p> <p><u>Example:</u></p> <p><u>Command Line</u></p> <p><u>Workaround</u> Install latest GHS patch from here</p>
No. a132	Introduction of Multi V7.1.6/2020.1.5
	<p><u>Version Information</u> V7.1.6/2020.1.5</p> <p><u>Details:</u> MULTI 7 is a major release</p> <ul style="list-style-type: none"> • A modern, multi-core, 64-bit x86 processor. Supports one of the following: <ul style="list-style-type: none"> ○ Windows 8 (64-bit). ○ Windows 10 (64-bit). • The compiler now supports C11, C17, and C18 • Discretionary Errors Now Given in C++ Standard Library Headers • Improved Download to RAM Dialogue in Multi Debugger <p><u>Example:</u></p> <p><u>Command Line</u></p> <p><u>Workaround</u></p>

(D) Valid Specification**Multi2000 V6.1.4/2013.5.5:**

Item	Date published	Document No.	Document Title
1	April 24, 2012	start-463286	Getting Started
2	October 4, 2013	build_v800-496213	Building Applications for Embedded V800
3	April 24, 2012	edit-463286	Editing Files and Configuring the IDE
4	October 4, 2013	connect_v800-496213	Configuring Connections for V800 Targets
5	August 29, 2012	debug-471430	Debugging
6	April 24, 2012	debug_cmd-463286	Debugging Command reference

Multi2000 V6.1.6/2015.1.7:

Item	Date published	Document No.	Document Title
1	April 24, 2014	start-506382	Getting Started
2	October 21, 2015	build_v800-544152	Building Applications for Embedded V800
3	April 24, 2014	edit-506382	Editing Files and Configuring the IDE
4	October 21, 2015	connect_v800-544152	Configuring Connections for V800 Targets
5	February 19, 2015	debug-525957	Debugging
6	April 24, 2014	debug_cmd-506382	Debugging Command reference

Multi2000 V7.1.4/2017.1.5:

Item	Date published	Document No.	Document Title
1	April 24, 2014	start-506382	Getting Started
2	October 21, 2015	build_v800-544152	Building Applications for Embedded V800
3	April 24, 2014	edit-506382	Editing Files and Configuring the IDE
4	October 21, 2015	connect_v800-544152	Configuring Connections for V800 Targets
5	February 19, 2015	debug-525957	Debugging
6	April 24, 2014	debug_cmd-506382	Debugging Command reference

Multi2000 V7.1.6/2018.1.5:

Item	Date published	Document No.	Document Title
1	May 8, 2017	start-592071	Getting Started
2	January 22, 2018	build_v800-612119	Building Applications for Embedded V800
3	May 8, 2017	edit-592071	Editing Files and Configuring the IDE
4	January 22, 2018	connect_v800-612119	Configuring Connections for V800 Targets
5	May 8, 2017	debug-592071	Debugging
6	May 8, 2017	debug_cmd-592071	Debugging Command reference

Multi2000 V7.1.6/2019.1.5:

Item	Date published	Document No.	Document Title
1	May 8, 2017	start-592071	Getting Started
2	January 24, 2019	build_v800-640037	Building Applications for Embedded V800
3	May 8, 2017	edit-592071	Editing Files and Configuring the IDE
4	January 24, 2019	connect_v800-640037	Configuring Connections for V800 Targets
5	May 8, 2017	debug-592071	Debugging
6	May 8, 2017	debug_cmd-592071	Debugging Command reference

Multi2000 V7.1.6/2020.1.5:

Item	Date published	Document No.	Document Title
1	May 8, 2017	start-592071	Getting Started
2	April 24, 2020	build_v800-671823	Building Applications for Embedded V800
3	May 8, 2017	edit-592071	Editing Files and Configuring the IDE
4	April 24, 2020	connect_v800-671823	Configuring Connections for V800 Targets
5	May 8, 2017	debug-592071	Debugging
6	May 8, 2017	debug_cmd-592071	Debugging Command reference

(E) Revision History

Item	Date published	Document No.	Comment
43	12-11-2013	r20tu0003ed1809	Items a105*, a106 are added.
44	21-03-2014	r20tu0003ed1810	Items a107 , a108 , a109 and a110 are added.
45	29-04-2014	r20tu0003ed1811	Item a111 added, some general corrections in the document
46	16-09-2014	r20tu0003ed1812	Items a112 , a113 , a114 , a115 , a116 and a117 are added. Change in patch release procedure: Original GHS report numbers are now referenced.
47	28-08-2015	r20tu0003ed1813	Item a118 and a119 added
48	09-09-2015	r20tu0003ed1814	Item a120 added
49	16-11-2015	r20tu0003ed1815	Item a121 added
50	23-11-2015	r20tu0003ed1816	Item a122 added, Item a121 modified.
51	12-01-2016	r20tu0003ed1817	Improved description of item a107
52	07-04-2016	r20tu0003ed1817	Item a123 added.
53	22-04-2016	r20tu0003ed1818	Corrected entries in the Table Of Operation Precautions, changed description of Item a123
54	16-11-2016	r20tu0003ed1819	Items a124 , a125 and a126 added
55	08-08-2017	r20tu0003ed1820	Item a127 added: Product Update MULTI 7.1.4 compiler 2017.1.5 MULTI 3.x and MULTI4.x information removed
56	01-12-2017	r20tu0003ed1821	Items a128 added
57	17-09-2018	r20tu0003ed1822	Items a129 , a130 and a131 added. Additional tool chain V7.1.6/2018.1.5 now included, added installation hints of Green Hills gpatch utility.
58	12-01-2021	r20tu0003ed1823	New release of V2020.1.5 Customer Notification updated to support RH850 targeted GHS tool chain only (V2013.5 or later).

* Note: item 43, a105 concerns Multi version release prior to V2013.5.5

