

# Release Notes for C51

## 8051 Development Tool Kits

This file contains release notes and last minute changes.

Information in this file, the accompany manuals, and software is  
Copyright © 2012 ARM Ltd and ARM Germany GmbH.  
All rights reserved.

---

## Contents

1. [What's New in C51](#)
2. [Example Programs](#)
3. [Device Database™](#)
4. [Peripheral Simulation](#)
5. [Technical Support](#)
6. [Contact Details](#)

## What's New in C51

The following sections list the changes instituted in each release of the C51 toolset.

### C51 Version 9.51 Release

- **[New Supported Devices]**
  - **[Silicon Laboratories Inc.](#)**  
[C8051F370](#), [C8051F371](#), [C8051F374](#), [C8051F375](#), [C8051F390](#), [C8051F391](#), [C8051F392](#), [C8051F393](#), [C8051F394](#), [C8051F395](#), [C8051F396](#), [C8051F397](#), [C8051F398](#), [C8051F399](#), [C8051T620](#), [C8051T621](#), [C8051T622](#), [C8051T623](#), [C8051T626](#), and [C8051T627](#).
- **[C51 Compiler]**
  - Corrected: calculation of negative constants within nested calls may create incorrect results (this problem was introduced in C51 V9.50a). For example:

```
#define TDO 5
#define GET_TDO() (Arr[TDO])

unsigned char xdata Arr[10];

unsigned char TestTDO() {
    unsigned char ret;
    ret = (unsigned char)((GET_TDO()-1)*2)-1);    // Incorrect result. For the negative constant a subtraction has been used instead of an addition.
```

```
    return ret;
}
```

- Corrected: incorrect pointer arithmetic with subtract of unsigned int variables for XDATA arrays with sizeof < 256 bytes. For example:

```
unsigned char xdata b[256]; /* Problem does not exist when array size > 256 */
void xdata *p;
unsigned int i = 256;      /* Problem only appears for unsigned int variables */

void main (void) {
    p = &b[256-i];          /* Works no problem when array index is used */
    p = b + 256 - i;        /* Failed on pointer arithmetic when uint variable is subtracted */
}
```

- **[LX51 Linker/Locator]**

- Corrected: a potential DPTR corruption which may occurs in code-banking applications when Global Register Coloring is enabled.

- **[[µVision](#)]**

- This C51 release comes with **µVision V4.60.6.8**.
- Enhanced: the Logic Analyzer allows rearranging signals through a simple drag&drop of the signal name. Signals can be scaled in width and height.
- Corrected: under certain circumstances the Source Browser incorrectly reported several definitions of an "enum".
- Refer to [uVision Revision History](#) for a complete list.

## C51 Version 9.50a Release

- **[[µVision](#)]**

- This C51 release comes with **µVision V4.53.0.6**.

## C51 Version 9.50 Release

- **[New Supported Devices]**

- **[Cypress](#)**

[CY8C3244AXI-153](#), [CY8C3244LTI-123](#), [CY8C3244LTI-130](#), [CY8C3244PVI-133](#), [CY8C3245AXI-158](#), [CY8C3245AXI-166](#), [CY8C3245LTI-129](#), [CY8C3245LTI-139](#), [CY8C3245LTI-144](#), [CY8C3245LTI-163](#), [CY8C3245PVI-134](#), [CY8C3245PVI-150](#), [CY8C3246AXI-131](#), [CY8C3246AXI-138](#), [CY8C3246LTI-125](#), [CY8C3246LTI-128](#), [CY8C3246LTI-149](#), [CY8C3246LTI-162](#), [CY8C3246PVI-122](#), [CY8C3246PVI-147](#), [CY8C3444AXI-116](#), [CY8C3444LTI-110](#), [CY8C3444LTI-119](#), [CY8C3444PVI-100](#), [CY8C3445AXI-104](#), [CY8C3445AXI-108](#), [CY8C3445LTI-078](#), [CY8C3445LTI-079](#), [CY8C3445LTI-081](#), [CY8C3445LTI-089](#), [CY8C3445PVI-090](#), [CY8C3445PVI-094](#), [CY8C3446AXI-099](#), [CY8C3446AXI-105](#), [CY8C3446LTI-073](#), [CY8C3446LTI-074](#), [CY8C3446LTI-083](#), [CY8C3446LTI-085](#), [CY8C3446PVI-076](#), [CY8C3446PVI-091](#), [CY8C3446PVI-102](#), [CY8C3665AXI-010](#), [CY8C3665AXI-016](#), [CY8C3665LTI-006](#), [CY8C3665LTI-044](#), [CY8C3665PVI-007](#), [CY8C3665PVI-008](#), [CY8C3665PVI-080](#), [CY8C3666AXI-036](#), [CY8C3666AXI-037](#), [CY8C3666AXI-052](#), [CY8C3666LTI-027](#), [CY8C3666LTI-050](#), [CY8C3865AXI-019](#), [CY8C3865LTI-014](#), [CY8C3865LTI-062](#), [CY8C3865PVI-060](#), [CY8C3865PVI-063](#), [CY8C3866AXI-035](#), [CY8C3866AXI-039](#), [CY8C3866AXI-040](#), [CY8C3866LTI-030](#), [CY8C3866LTI-067](#), [CY8C3866LTI-068](#), [CY8C3866PVI-021](#), and [CY8C3866PVI-070](#).

- **Zilog**  
[Z51F0410](#), [Z51F3220](#), [Z51F3221](#), [Z51F6412](#), and [Z51F811](#).
- **[µVision]**
  - This C51 release comes with **µVision V4.53.0.4** which includes the new **Scintilla** based editor.
  - The new editor includes the following **enhancements**:  
Encoding for **UTF-8 Unicode**, **DBCS Korean**, **DBCS Japanese**, and **DBCS Chinese** languages. **Unicode** and Asian **ANSI** encoding is recognized automatically when a file is opened.  
Monospaced fonts and proportional fonts are supported.  
Syntax coloring has been extended.  
Unprintable characters, such as End-Of-Line, can be visualized in the editor.  
The **Outlining** menu has been simplified. Outlining information is saved and restored for each file.  
**Search and replace** utilities (Incremental Find, Find-in-Files, and Replace) have been reworked.  
Text can be zoomed with **Ctrl+mouse wheel**. The information is saved and restored for each file.  
In case device-specific books are not found in the local installation, then [www.keil.com](http://www.keil.com) is scanned for a matching document.
  - and **corrections**: Scrolling quickly through large files with **Page Up** or **Page Down** works smoothly.  
The editor's context menu can be closed by pressing **ESC**.  
Breakpoints can be set now with a simple click into the editor margin.  
Under some circumstances the Debugger showed wrong values of arrays or structures in the Watch window.
  - Refer to [Revision History](#) for a complete list.

## C51 Version 9.06 Release

- **[New Supported Devices]**
  - **Atmel** [AT89LP51RB2](#), [AT89LP51RC2](#), [AT89LP51IC2](#), [AT89LP51RD2](#), [AT89LP51ED2](#), and [AT89LP51ID2](#) devices.
  - **Infineon** [TLE9831](#), [TLE9832](#), [TLE9833](#), [TLE9834](#), and [TLE9835](#) devices.
  - **Maxim** [71M6531D](#), [71M6531E](#), [71M6532D](#), [71M6532F](#), [71M6534H](#), [71M6541D](#), [71M6541E](#), [71M6542E](#), [71M6543F](#), and [71M6543G](#) devices.
  - **Silabs** [C8051F969](#), [C8051F968](#), [C8051F967](#), [C8051F966](#), [C8051F965](#), [C8051F964](#), [C8051F963](#), [C8051F962](#), [C8051F961](#), [C8051F960](#), [Si1037](#), [Si1036](#), [Si1035](#), [Si1034](#), [Si1033](#), [Si1032](#), [Si1031](#), [Si1030](#), [Si1027](#), [Si1026](#), [Si1025](#), [Si1024](#), [Si1023](#), [Si1022](#), [Si1021](#), and [Si1020](#) devices.
- **[Device Support]**
  - Added: Quick\_Test example for **Infineon TLE983x** based devices in folder **..\C51\Examples\Infineon TLE983x\**.
  - Added: banking example for **Maxim 71M6543G** device in folder **..\C51\Examples\Maxim\BankEx1\**.
  - Added: banking example for **Maxim 71M6534H** device in folder **..\C51\Examples\Maxim\BankEx2\**.
- **[µVision]**
  - This C51 release comes with µVision V4.24.00.

## C51 Version 9.05 Release

### ▪ [Cx51 Compiler]

- Improved: access to bit-field members with size 1 bit. The compiler uses bit instructions to access such bit-field members objects that are defined with the **bdata** memory type. This is now extended also to structs that are defined with the extern attribute.
- Corrected: Common sub-expression elimination can deliver incorrect values when array pointers are used. Example:

```
int foo (unsigned char dat[]) {
    int len1, len2, ofs;

    ofs = 5;
    len1 = dat[ofs];
    if(len1 > 0x10) return -1;
    ofs += len1 + 1;           // modify 'ofs'
    len2 = dat[ofs];           // 'dat[ofs]' not reloaded, instead value of 'len1' is used
    return len2;
}
```

- Corrected: Wrong code with pointer arithmetic and conversions to long. Example:

```
unsigned char *p;
unsigned int code a1[10];
unsigned char xdata a2[500];

void foo (void) {
    unsigned long r1 = (unsigned long)(p - (a2 + a1[0]));           // wrong
    unsigned long r2 = (unsigned long)(p - (unsigned long)(a2 + a1[0])); // work around
}
```

- Corrected: Pointer arithmetic with conversion to 'unsigned long' rejected. Example:

```
unsigned char xdata *ptr1;
unsigned char xdata *ptr2;
unsigned long i4;

void foo (void) {
    i4 = (ptr1 - ptr2); // pointer conversion rejected, instead an error was issued
}
```

### ▪ [LX51 Linker/Locater]

- Corrected: When Linker Code Packing is used with a banking application, segments or the content of segments may get lost. This problem was introduced with C51 Version 9.03.

### ▪ [New Supported Devices]

- **CoreRiver** [ADCore200](#), [ADCore210](#), [ADCore220](#), [AmpCore100](#), [ChargerCore2.0](#), [GC230](#), [GC400](#), [GC410](#), [GC81L541A0](#), [GC81L581A0](#), [GC81L591A0](#), [GC89L541A0](#), [GC89L581A0](#), [GC89L591A0](#), and [HallCore110](#) devices.
- **Nordic** [nRF8200](#) device.

- **Nuvoton** [N78E055A](#), [N78E059A](#), [N78E366A](#), [N78E517A](#), [N79E234](#), [N79E235](#), [N79E342](#), [N79E352](#), [N79E822](#), [N79E823](#), [N79E824](#), [N79E825](#), [N79E843](#), [N79E844](#), [N79E845](#), [N79E853](#), [N79E854](#), [N79E855](#), [N79E875](#), [W78E051D](#), [W78E052D](#), [W78E054D](#), [W78E058D](#), and [W78E516D](#) devices.
- **Silabs** [C8051T620](#), [C8051T621](#), [C8051T622](#), [C8051T623](#), [C8051T320](#), [C8051T321](#), [C8051T322](#), [C8051T323](#), [C8051T326](#), [C8051T327](#), [C8051F380](#), [C8051F381](#), [C8051F382](#), [C8051F383](#), [C8051F384](#), [C8051F385](#), [C8051F386](#), and [C8051F387](#), devices.
- **[µVision]**
  - This C51 release comes with µVision V4.22.00.

## C51 Version 9.03 Release

- **[Cx51 Compiler]**
  - Improved: Evatronix R8051XC2 core now uses the MDU for signed long divisions.
  - Corrected: With OPTIMIZE(8, SIZE) or higher optimization levels there is a potential problem with common code generation of long/float store operations that store constants with different memory types. Example:

```
int flag;
long xdata l;
long *pl;

void main (void) {
    if (flag) {    l = 0; flag = 1; }    // store with identical constant value but different space
    else        { *pl = 0; flag = 1; }    // may create incorrect common blocks
}
```
- **[LX51 Linker/Locater]**
  - Improved: When using Linker Code Packing the gaps in the BANKAREA are not optimized.
  - Corrected: Potential SEGMENT OVERLAPS when using Linker Code Packing on a banking application.
- **[New Supported Devices]**
  - **Atmel** [AT80C51RD2](#), [AT89LP213](#), [AT89LP216](#), [AT89LP52](#), and [AT89LP6440](#) devices.
  - **Infineon** [XC878LM-13F](#), [XC878CLM-13F](#), [XC878LM-16F](#), [XC878CLM-16F](#), [XC874-13F](#), [XC874LM-13F](#), [XC874CM-13F](#), [XC874CLM-13F](#), [XC874-16F](#), [XC874LM-16F](#), [XC874CM-16F](#), [XC874CLM-16F](#), and [XC836MT-1F](#) devices.
  - **NXP** [P87C51FA](#), [P87C51FB](#), [P89CV51RB2](#), [P89CV51RC2](#), [P89CV51RD2](#), [P89LPC779](#), and [P87V660X2](#) devices.
  - **Silabs** [C8051F348](#), [C8051F349](#), [C8051T606](#), [C8051T630](#), [C8051T631](#), [C8051T632](#), [C8051T633](#), [C8051T634](#), [C8051T635](#), [C8051F540](#), [C8051F541](#), [C8051F542](#), [C8051F543](#), [C8051F544](#), [C8051F545](#), [C8051F546](#), [C8051F547](#), [C8051F550](#), [C8051F551](#), [C8051F552](#), [C8051F553](#), [C8051F554](#), [C8051F555](#), [C8051F556](#), [C8051F557](#), [C8051F560](#), [C8051F561](#), [C8051F562](#), [C8051F563](#), [C8051F564](#), [C8051F565](#), [C8051F566](#), [C8051F567](#), [C8051F568](#), [C8051F569](#), [C8051F570](#), [C8051F571](#), [C8051F572](#), [C8051F573](#), [C8051F574](#), [C8051F575](#), [C8051F34A](#), [C8051F34B](#), [C8051F34C](#), [C8051F34D](#), [C8051F716](#), [C8051F717](#), [C8051F980](#), [C8051F981](#), [C8051F982](#), [C8051F983](#), [C8051F985](#), [C8051F986](#), [C8051F987](#), [C8051F988](#), [C8051F989](#), [C8051F990](#), [C8051F991](#), [C8051F996](#), [C8051F997](#), [C8051F800](#), [C8051F801](#), [C8051F802](#), [C8051F803](#), [C8051F804](#), [C8051F805](#), [C8051F806](#), [C8051F807](#), [C8051F808](#), [C8051F809](#), [C8051F810](#), [C8051F811](#), [C8051F812](#), [C8051F813](#), [C8051F814](#), [C8051F815](#), [C8051F816](#), [C8051F817](#), [C8051F818](#), [C8051F819](#), [C8051F820](#), [C8051F821](#), [C8051F822](#), [C8051F823](#), [C8051F824](#), [C8051F825](#), [C8051F826](#), [C8051F827](#), [C8051F828](#), [C8051F829](#), [C8051F830](#), [C8051F831](#), [C8051F832](#), [C8051F833](#), [C8051F834](#), [C8051F835](#), [C8051F901](#), [C8051F902](#), [C8051F911](#), [C8051F912](#), [Si1000](#), [Si1001](#), [Si1002](#), [Si1003](#), [Si1004](#), [Si1005](#), [Si1010](#), [Si1011](#), [Si1012](#), [Si1013](#), [Si1014](#), [Si1015](#), [C8051F520A](#), [C8051F521A](#), [C8051F523A](#), [C8051F524A](#), [C8051F526A](#), [C8051F527A](#), [C8051F530A](#), [C8051F531A](#), [C8051F533A](#), [C8051F534A](#), [C8051F536A](#), and [C8051F537A](#) devices.

- **Teridian Semiconductors** [78M6612](#), [78M6618](#), and [71M6543](#) devices.
- **Texas Instruments** [CC2530F32](#), [CC2530F64](#), [CC2530F128](#), [CC2530F256](#), [CC2531F128](#), [CC2531F256](#), [CC2533F32](#), [CC2533F64](#), [CC2533F96](#), [CC2540F128](#), and [CC2540F256](#) devices.
- **[µVision]**
  - This C51 release comes with µVision V4.14.16.

## C51 Version 9.02 Release

- **[C51 Compiler]**
  - Improved: access to bit-field members with size 1 bit. The compiler uses bit instructions to access such bit-field members. When objects are defined with the **bdata** memory type, direct bit addressing is used. Example:

```

struct bf { unsigned char b0:1; unsigned char b1:1; };
struct bf      a;
struct bf bdata b;
:
if (a.b0 && b.b1)  b.b1 = 0;
```
  - Corrected: multiplication `long = int * int` is potentially incorrect in Dallas 390 mode.
  - Corrected: explicit cast to unsigned char was ignored with complex address arithmetic. Example:

```

unsigned char far table[256];
unsigned char i, v;
:
v = table[(unsigned char)(16+i+20)];  // index now truncated to 8-bit
```
  - Corrected: when using conditional operators (`? :`) in complex parameter lists, there is a potential for unbalanced PUSH / POP instructions. This typically creates a application program crash at the function call.
- **[C Run-Time Library]**
  - Corrected: the function `toint` did not detect the value range 0x59 - 0x40 as invalid. Now the function returns -1 for these values.
  - Corrected: timing of Multiplication Division Unit (MDU) in Evatronix R8051XC2 device is faster and now reflected in the C Library. The MDU timing for int/long multiplication and long division is adjusted.
- **[New Supported Devices]**
  - **Infineon** [XC835MT-2F](#), [XC836-2F](#), [XC836M-1F](#), [XC836M-2F](#), [XC836MT-2F](#), and [XC836T-2F](#) devices.
- **[Device Simulation]**
  - Corrected: SiLabs C8051F41x devices: SMBus simulation when using I<sup>2</sup>C generator.
  - Corrected: SiLabs C8051F12x devices: automatic page switch for interrupts and timing of timer 2/3/4.
  - Corrected: SiLabs C8051F12x devices: on I<sup>2</sup>C the receive of more than 256 bytes now generates a stop.
  - Corrected: SiLabs C8051F36x devices: crossbar did not connect the right I/O signals under some circumstances.
  - Corrected: Evatronix T8051: CPU instruction timing.

## C51 Version 9.01 Release

- **[C51 Compiler]**
  - Corrected: when MODDA is used and *int* numbers are multiplied and assigned to *long*, the result is potential incorrect.
- **[Device Support]**
  - Added: debug support for **Infineon XC82X** devices.
- **[New Supported Devices]**
  - **Infineon** [XC822T-0F](#), [XC822M-1F](#), [XC822MT-1F](#), [XC824MT-1F](#), and [XC824M-1F](#) devices.

## C51 Version 9.00 Release

- **[µVision4]**
  - This C51 release comes with the new [µVision4 IDE](#).
- **[New Supported Devices]**
  - **Evatronix** [R8051XC\(1 DPTR\)](#), [R8051XC2\(1 DPTR\)](#), [R8051XC2\(2 DPTR\)](#), [R8051XC2\(8 DPTR\)](#), [R8051XC2-A\(1 DPTR\)](#), [R8051XC2-A\(2 DPTR\)](#), [R8051XC2-A\(8 DPTR\)](#), [R8051XC2-AF](#), [R8051XC2-B\(1 DPTR\)](#), [R8051XC2-B\(2 DPTR\)](#), [R8051XC2-B\(8 DPTR\)](#), and [R8051XC2-BF](#) devices.
  - **Nordic** [nRFLU1P-F16](#) and [nRFLU1P-F32](#) devices.
  - **NXP** [P89LPC9361](#) device.
  - **Silabs** [C8051F580](#), [C8051F581](#), [C8051F582](#), [C8051F583](#), [C8051F584](#), [C8051F585](#), [C8051F586](#), [C8051F587](#), [C8051F588](#), [C8051F589](#), [C8051F590](#), and [C8051F591](#) devices.
- **[Device Support]**
  - Corrected: device settings for **Infineon** [XC888-6FF](#), [XC888CM-8FF](#), [XC888LM-6FF](#), [XC886-6FF](#), [XC866L-1FR](#), [XC866L-2FR](#), [XC866L-4FR](#), and [XC864-1FRI](#) devices.
  - Corrected: device settings for **Nordic Semiconductor** [nRF24E1](#), [nRF24E2](#), [nRF9E5](#), [nRF24LU1](#), and [nRF24LE1](#) devices.
  - Added: debug support for **NXP** [P89LPC9361](#) and [P89LPC954](#) devices in the **LPC900 EPM** Emulator/Programmer.
  - Updated: **LPC900 EPM** flash programmer configuration dialog to support devices with 16 flash sectors.
- **[CX51 Compiler]**
  - Corrected: constant folding of two negative array index values. For example:

```
unsigned char arr[512];
unsigned int i;

i = arr[i-1-5];    // incorrect in C51 V8: arr[i-4] instead of arr[i-6]
```
  - Corrected: when using the NOAREGS directive, complex arithmetic with nested calls may create incorrect results. For example:

```
#pragma NOAREGS
```

```
int result;
extern char f(unsigned char idx);

result = (f(1)*0x100+f(0)) - (f(3)*0x100+f(2)); // incorrect result. POP destroys value in ACC
```

- **[LX51 Linker/Locater]**

- Corrected: when using OPTIMIZE(10) or above, there was a potential that common code blocks are called incorrectly. Therefore programs may have operated incorrectly.

- **[BL51 Linker/Locater]**

- Corrected: when using RTX51 user interrupt functions were overlapping with RTX ISR vectors which resulted in a linker warning.
- Corrected: data overlaying may not work when the last input module contains an interrupt function; the linker incorrectly issues WARNING 16: main uncalled.

## C51 Version 8.18 Release

- **[Device Support]**

- Added: debug support for **NXP** [P89LPC9408](#) in the **LPC900 EPM Emulator/Programmer**.

- **[New Supported Devices]**

- **Nuvoton** [W681308](#) device.
- **NXP** [P89LPC9201](#), [P89LPC9211](#), [P89LPC922A1](#), [P89LPC9241](#), [P89LPC9251](#), [P89LPC9301](#), [P89LPC931A1](#), [P89LPC9331](#), [P89LPC9341](#), and [P89LPC9351](#) devices.
- **Silabs** [C8051F500](#), [C8051F501](#), [C8051F504](#), [C8051F505](#), [C8051F506](#), [C8051F507](#), [C8051F508](#), [C8051F509](#), [C8051F510](#), [C8051F511](#), [C8051F700](#), [C8051F701](#), [C8051F702](#), [C8051F703](#), [C8051F704](#), [C8051F705](#), [C8051F706](#), [C8051F707](#), [C8051F708](#), [C8051F709](#), [C8051F710](#), [C8051F711](#), [C8051F712](#), [C8051F713](#), [C8051F714](#), and [C8051F715](#) devices.

- **[ULINK2 Support]**

- Corrected: potential deadlock on ST uPSD targets with ULINK2 solved.

- **[Device Simulation]**

- Corrected: **Infineon** XC800 simulation of the MDU was incorrectly implemented.
- Corrected: For C8051F12x/F13x devices the EXFn and TOGn behavior matches with latest information from **SiLABS**.
- Added: simulation for **Atmel** [AT89C51RE2](#). The second UART is now available for the simulation.

- **[CX51 Compiler]**

- Corrected: initialization failed on far addresses when the object is located with `_at_`. For example:

```
#pragma VARBANKING 02

unsigned char far array[65530] _at_ 0x150006;
unsigned long x1 = (unsigned long)(array); // incorrect address stored in 'x1'.
```



## C51 Version 8.17a Release

### ▪ [Device Support]

- Added: debug support for the follow devices from **Analog Devices** [ADE5166](#), [ADE5169](#), [ADE5566](#), [ADE5569](#), [ADE7166F16](#), [ADE7166F8](#), [ADE169F16](#), [ADE7566F16](#), and [ADE7566F8](#) in the **ADI Monitor Driver**.
- Added: debug support for **NXP** P89LPC9321 and P89LPC9351 devices in the **LPC900 EPM Emulator/Programmer**.

### ▪ [Device Support]

- Added: Nuvoton devices in the device database.
- Corrected: Port Pin P2.2 is available (instead of P2.5) on **NXP P89LPC917** devices. This is now reflected in the peripheral dialogs.
- Corrected: UART0 baudrate is now correctly displayed when Timer2/3/4 is used as baudrate generator on **SiLabs C8051F13x** devices.

### ▪ [Device Simulation]

- Corrected: simulation of Reset Source Register (RSTSRC) and SFR Page Control Register (SFRPGCN) for **SiLabs C8051Fxxx** devices.
- Corrected: handling of Automatic Page Control Enable (SFRPGCN) and Reset Source Register (RSTSRC) for **SiLabs C8051Fxxx** devices.
- Corrected: simulation of PLLLCK (PLL Lock Flag) for **SiLabs C8051F12x/13x** devices. PLLLCK is now set when PLL is configured correctly and frequency is locked.
- Added: support for **V**: user-defined memory area for **NXP 80C51MX** devices.
- Corrected: simulation issues with the Evatronix R8051XC peripherals DMA and interrupt.

### ▪ [CX51 Compiler]

- Corrected: when two long operands are loaded from complex arrays (each with object size bigger than 256 bytes), there was a potential register overwrite in register R0. The result of the long operation was in such cases incorrect.

Example:

```
struct s2 { unsigned int idx : 1; } s2;
struct s { unsigned long l1; unsigned char a[256]; unsigned long l2; } xdata sarr[2];

unsigned long l;

void main (void) {
    l = sarr[s2.idx].l1 + sarr[s2.idx].l2; // incorrect result of long addition
}
```

### ▪ [AX51 Macro Assembler]

- Corrected: in NXP 80C51MX mode, DATA, IDATA, and EDATA can be placed to absolute addresses 0x7F0000 and above. This is now accepted.
- Added: [ECRM directive](#) that allows to expand generic CALL instructions to ECALL for NXP 80C51MX devices.

## C51 Version 8.16a Release

### ▪ [CX51 Compiler]

- Corrected: when *int* numbers are multiplied and assigned to *long*, the result is potential incorrect. This problem has been introduced in V8.15.
- Corrected: C51 `_at_` problem with linker code packing fixed.
- **[AX51 Macro Assembler]**
  - Enhanced: for the NXP MX devices, **CALL/JMP** instructions are encoded to **ECALL/JMP** when needed..
- **[Device Support]**
  - Added: ULINK and Infineon DAS (Device Access Server) support for the XC864 device.
  - Enhanced: Evatronix R8051XC XDATA Banking example optimized.
  - Enhanced: Infineon XC800 startup code.
  - Added: Support for Infineon XC864 including a Blinky example.
  - Added: Syntek Semiconductors [STK6031](#) and [STK6032](#) devices to device database.
- **[Device Simulation]**
  - Added: for SiLABS [C8051F360/1/2/3/4/5/6/7/8/9](#) and [C8051F410/1/2/3](#).

## C51 Version 8.15 Release

- **[Cx51 Compiler]**
  - Corrected: interrupt functions combined with [NOINTVECTOR](#) where not detected by the linker as new root and an incorrect linker warning was reported.
  - Corrected: when using Dallas 390 mode with ROM(D512K) or ROM(D16M), pdata arrays could not be located anywhere in memory.
  - Corrected: when using the XCROM directive in combination with function pointers, constant initializations where omitted.
  - Enhanced: long multiplication with two unsigned int/char arguments has now a much higher performance.
- **[Target Support]**
  - Added: support for the Infineon USCALE XC800 hardware via the **Infineon DAS Client for XC800**.
- **[Device Simulation]**
  - Corrected: access to MACACC was not corrected simulated for SiLABS [C8051F12x](#) and [C8051F13x](#) devices.
  - Added: device support and simulation for Infineon [XC878](#).
  - Added: simulation for new peripherals (Software Reset, RTC, DMA) on [Evatronix R8051XC](#) core.
  - Added: xdata banking support for [Evatronix R8051XC](#) core.
- **[LX51 Linker/Locater]**
  - Corrected: Linker Code Packing may combine incorrectly blocks from several code banks into common areas.
- **[ULINK2 Support]**
  - Added: Debug and Flash-Programming support for NXP [P89LPC952](#) and [P89LPC954](#).

## C51 Version 8.12 Release

- **[Device Simulation]**
  - Added: device support and simulation for SiLABS [C8051T600/1/2/3/4/5](#) and [C8051T610/1/2/3/4/5/6/7](#).
- **[Cx51 Compiler]**
  - Corrected: nested call with struct pointer arguments where incorrectly processed.
- **[LX51 Linker/Locator]**
  - Corrected: sfr16 definitions in assembly code and C source file may generate Warning L46: SFR SYMBOL HAS DIFFERENT VALUE.

## C51 Version 8.11a Release

- **[Device Support]**
  - Added: support for Ramtron [VRS51L3072](#) and [VRS51L3174](#).
  - Added: support for Nordic Semiconductor [nRF24LU1](#).
- **[Device Simulation]**
  - Added: device support and simulation for SiLABS [C8051F336](#), [C8051F337](#), [C8051F338](#), and [C8051F339](#).
  - Corrected: PORTx and PCA output pins on SiLABS C8051F12x did not correctly update in simulation.

## C51 Version 8.10 Release

- **[Device Support]**
  - Added: support for Megawin [MPC82G516A](#) and [MPC82L54A](#).
  - Enhanced: startup code for Infineon XC800 devices has now selections for device variants. Infineon XC88x AC step devices requires to set the device to VCO bypass mode before PLL switching.
  - Added: support for NXP P89V52X2.
- **[Device Simulation]**
  - Corrected: simulation of MULRDY and OSCICL corrected for SiLABS C8051F3xx series.
  - Enhanced: simulation of Evatronix R8051XC watchdog timer with optional prescaler. For details refer to [Application Note 191: Toolchain Extensions for R8051XC Core](#).
  - Corrected: DPTR simulation of Evatronix R8051XC; when 2 DPTR where selected the auto-increment feature (DPC register) did not work. Two R8051XC devices are now in the device database: **R8051XC (8 DPTR)** with simulation for 8 DPTR, **R8051XC (2 DPTR)** with simulation for 2 DPTR.
  - Corrected: SiLABS simulation for UART #1 had a problem with the transmit interrupt bit (TI) when SFR page was set to 1.
- **[Cx51 Compiler]**
  - Corrected: the MODC2 directive did not correctly save and restore multiple DPTR registers on interrupt entry/exit.
  - Corrected: Assembler instructions inserted with #pragma ASM trigger now register usage of all CPU registers and therefore avoids register clashes.
  - Corrected: strcmp and strncmp library functions changed from signed char to unsigned char compare (ANSI requirement).

- Improved: detection of conflicting memory types when used in combination with typedef's, for example:

```
typedef char    code CCHAR;
typedef CCHAR xdata XCHAR; // generates now WARNING C185: different memory space
CCHAR    idata var2;       // generates now WARNING C185: different memory space
```

- **[LX51 Linker/Locater]**

- Corrected: REMOVEDUNUSED did not correctly work with SROM symbols and linker code packing.
- Corrected: debug symbols of absolute bits generated in AX51 had wrong offset.
- Corrected: segment locating with the LAST keyword generated unnecessary memory gaps when used with code banking.
- Corrected: when linker code packing is used, an address reference to *far* variables with *\_at* placement was wrong.

- **[BL51 Linker/Locater]**

- Corrected: segments with a AJMP instruction as last instruction where located at the end of a 2KB block which generated a linker error.

- **[ULINK and ULINK2 Support]**

- Corrected: Verify failed on µPSD devices when common segments where located to code banks, but no bank 0 exists.

## C51 Version 8.09 Release

- **[Device Simulation]**

- Corrected: wrong timing of the Timer 1 in the Infineon XC800 devices.
- Corrected: clock calculation of Infineon XC88x devices was incorrect.
- Corrected: Dallas D80C400 simulation did not correctly switch to contiguous mode.
- Added: Simulation for [Atmel AT89C51AC3](#).

- **[Device Support]**

- Corrected: when debugging Infineon XC800 Devices with the DAS interface data/idata variables where shown incorrectly.

- **[Target Debugging for Analog Devices ADuC83x, ADuC84x, and ADE7xxx]**

**Options - Project - Debug - Use: ADI Monitor Driver** selects target driver for the Analog Devices ADuC834, ADuC84x, and ADE7xxx devices. This driver is now extended to support the 1-Pin Pod interface and the new ADE7xxx devices.

- **[Monitor-51, Monitor-390]**

- Corrected: potential communication problems with low-cost USB-COM adapters.

- **[Cx51 Compiler]**

- Corrected: a code generation problem with bit-field arrays when the array index is a function return value.

## C51 Version 8.08a Release

- **[Device Simulation]**

- Corrected: the previous version had a delay when starting signal functions. This delay is now removed so that the startup behavior is identical to releases before version 8.06.
- Corrected: the behavior of JBC instructions on I/O ports was not fully correct. JBC instructions now read the SFR register (Px) instead of the I/O port value (PORTx).
- **[Cx51 Compiler]**
  - Corrected: integer promotion was missing on complex arithmetic with char/unsigned char and multiplication or division.
- **[Ax51 Macro Assembler]**
  - Enhanced: the `_DATE2_` macro is now defined also for the A51 and AX51 Macro Assembler.
- **[uVision3 IDE]**
  - Corrected: potential crash in the project window on right mouse click when no item was selected.
- **[Device Support]**
  - Corrected: Peripheral display for Port 4 and Port 5 of the NXP 89LPC952 device was missing.
  - Enhanced: on XC800 devices the dialog **Project - Options - Debug - ULINK Settings - Disable interrupts during steps** is implemented. This option disables interrupts during single-stepping and therefore executes only instructions from the current function.
  - Added support for Infineon TLE78xx series.
  - Corrected: simulation for external interrupt inputs EINT0 and EINT1 on Infineon XC800 devices.
  - Corrected: debugger startup problems with the Infineon DAS server.

## C51 Version 8.06 Release

- **[Device Support]**
  - Infineon XC800 Devices: added debugging and flash programming support for new Infineon XC800 devices (XC866-1FR, XC856) with ULINK and Infineon DAS server.
  - Enhanced: Infineon XC800 startup code **START\_XC.A51** V1.02 that supports pdata addressing ([C51: USING PDATA VARIABLES ON INFINEON XC800](#)).
  - SST SmartCards: added core features for device simulation.
- **[Device Simulation]**
  - Corrected: interrupt vector for 2nd UART on Philips P89LPC952 was incorrectly configured in simulator and target dialog.
  - Enhanced: VTREG PPAGE is initialized to 0 on all Philips LPC900 devices to allow simulation of MOVX @Ri without configuration.
  - Corrected: a problem with AT89S8252 EEPROM simulation.
  - Corrected: a problem with the baudrate generation on Atmel devices with X2 feature (the baudrate was displayed incorrect).
  - Corrected: a problem with code banking on Mentor M8051EW core.
- **[Cx51 Compiler]**
  - Corrected: code that is not used (i.e. with macros) is removed, even when it creates other side effects.

- Corrected: unexpected error message for syntactical correct statement.

```
struct st2 { unsigned char uc1; unsigned char uc2; };
struct st1 { struct st2 st2;    unsigned char u1;    unsigned char u2;    };
struct st1 st;
```

```
void main (void) {
    (&st.st2)->uc1 = 0; // gives error, but should be Ok.
}
```

- Corrected: library function **toint** did no flag values 0x3A - 0x40 as incorrect.

- **[ULINK and ULINK2 Support]**

- Added: support for ULINK2 for the Infineon XC800 and ST uPSD series.
- Added: device support for Infineon XC886, XC888, and XC856.

## C51 Version 8.05 Release

- **[Device Simulation for Infineon XC88x Series]**

Added device simulation for Infineon XC886 and Infineon XC888.

- **[Device Simulation for Philips P89LPC952/954]**

Corrected: the two serial windows did not work for the Philips P89LPC952/954 devices.

- **[Device Simulation for M8051EW]**

Corrected: when M8051EW features were enabled, it was not possible to debug non-banking applications.

- **[Device Simulation for R8051XC]**

Enhanced: some features were not covered in the first revision of the simulator. Now the simulator also takes care about peripheral timing, and write operations to code memory.

- **[LX51 Linker/Locater]**

Corrected: when using interbank call table optimization (?B\_RST\_BANK != 0xFF), the linker was optimizing too much.

- **[ULINK Driver for ST uPSD]**

Corrected: flash programming did not work for uPSD3422 devices.

- **[Cx51 Compiler]**

- Enhanced: C51 did issue an error when accessing struct members with *s->member* instead of *s.member*.
- Corrected: using multiple dummy assignments to remove unused variables may cause register overwrites.
- Corrected: potential code problems when incrementing far pointers with long constants.

```
void func (void) {
    unsigned int i;
    long far* entry;

    while(i) {
        i--;           // increment missing
    }
}
```

```

    entry += 1L;    // due to 'far' pointer increment with 1L
}
}

```

- Corrected: potential problem on SmartMX, when using Optimize level 8 or 9 without OBJECTADVANCED.

```

#pragma MXP
int  xdata x;
char xdata * xdata p;

void f1 (char *);
void f2 (char *);

void main (void) {
    if (x == 0xFFFF) f1 (p);    // uses CMPW instruction that affects 'Z' flag
    else             f2 (p);    // MOVX A,Rx,@DPTR destroys 'Z' flag
                                // and cannot be combined
}

```

## C51 Version 8.04 Release

### ▪ [Cx51 Compiler]

- Enhanced: The *Warning C259: pointer: different mspace* is improved and covers now also situations where an address value is assigned.
- New: The *Warning C289: converting non-pointer to pointer* is issued when a integer value is assigned to a pointer.
- Corrected: A problem with the SRC file output is wrong with optimized address values.
- Corrected: dummy read to MD3 was missing for int\*int multiplication when using the Infineon/Evatronix MDU.
- Added: Support for the SiLABS Arithmetic Accelerator (available in C8051F12x and F13x devices) in the far banking library.

### ▪ [Device Support for Evatronix/Cast R8051XC Core]

Added complete simulation and compiler support for the features of the R8051XC core. Detailed information is provided in [Application Note 191: Toolchain Extensions for the R8051XC Core](#).

### ▪ [Device Support for Mentor M8051EW Core]

Enhanced simulation support for the Mentor M8051EW memory extension features. It is now possible to simulate code banking and far memory applications and the address of the extension core features of the R8051XC core. Detailed information is provided in [Application Note 171: Using M8051EW Memory Extension](#).

### ▪ [Device Support for Infineon XC800 Series]

Added ULINK debugging support and DAS driver for new Infineon XC800 devices (XC886, XC888). Detailed information is

### ▪ [BL51 and LX51 Linker/Locater]

Corrected: there might be incorrect *WARNING L15: MULTIPLE CALL TO FUNCTION* messages when using syntax: **OVERLAY (\* ! (func1, func2,func3,...))** to group more than just two functions.

## C51 Version 8.02 Release

### ▪ [uVision Debugger]

Added peripheral simulation support and target debugging dialogs for the following devices:

- Atmel [AT89C51RE2](#), [AT89C51IE2](#), [AT89C51CC03](#), [AT8xC51SND1](#), [AT89S8253](#), [AT89LP2052](#), [AT89LP4052](#)
- Revised: Atmel [AT89C5131/AT89C5131A](#) (added TWI), [AT8xC5132](#) (added TWI, ADC)
- Philips [P89LPC9102](#), [P89LPC9103](#), [P89LPC9107](#), [P89LPC9221](#), [P89LPC9311](#), [P89LPC932A1](#), [P89LPC938](#),
- Philips [P89LPC9401/P89LPC9408](#) (without LCD Driver), [P89LPC952](#), [P89LPC964](#), [P89LPC966](#)
- Philips [P89V660](#), [P90V662](#), [P89V664](#)
- **[LX51 Linker/Locater]**  
Corrected a problem with code banking, the linker reported incorrectly error L124: BANK SWITCH MODULE INCORRECT.
- **[Cx51 Compiler]**  
Corrected a macro expansion problem, array index calculations with negative offset, and far access to absolute memory locations.
- **[Target Debugging Support for Infineon XC800 Series]**  
Added ULINK debugging support and DAS driver for new Infineon XC800 devices (XC886, XC888).
- **[ULINK Driver for ST uPSD]**  
Added new JTAG device ID's for ST uPSD3212, uPSD3312, and uPSD3422.

## C51 Version 8.01 Release

- **[uVision Debugger]**  
Added peripheral simulation support and target debugging dialogs for the following devices:
  - [Silicon Labs C8051F120](#), [Silicon Labs C8051F121](#), [Silicon Labs C8051F122](#), [Silicon Labs C8051F123](#),
  - [Silicon Labs C8051F124](#), [Silicon Labs C8051F125](#), [Silicon Labs C8051F126](#), [Silicon Labs C8051F127](#),
  - [Silicon Labs C8051F130](#), [Silicon Labs C8051F131](#), [Silicon Labs C8051F132](#), [Silicon Labs C8051F133](#)
- **[P89LPC952 and MCB950 Board Support]**  
Added support for Philips P89LPC95x device series.
- **[AX51 Macro Assembler]**  
Corrected a problem with forward references in symbols which generates an error message.
- **[LX51 Linker/Locater - Code Packing]**  
Corrected a potential problem with code packing when modules are translated with different optimization levels (less than 8) but the OBJECTADVANCED directive.
- **[L51\_BANK.A51 - Code Banking Configuration File]**  
Changed behavior of the ?B\_RESTORE\_BANK entry with ?B\_MODE=0 to avoid glitches when using the RTX51 or RTX51 Tiny operating system together with code banking.
- **[Cx51 Compiler]**  
Version number changed for logistic reasons. No other changes compared to V8.00.



## C51 Version 8.00 Release

- **[uVision3 IDE]**  
The µVision3 IDE contains several enhancements orrected: the device simulation covers now the latest timing specification for Dallas DS89C420, DS89C430, DS89C440, and DS89C450 devices. The timer simulation was previously based on a older data book and therefore incorrect.
- **[Device Simulation]**
  - Corrected: the device simulation covers now the latest timing specification for Dallas DS89C420, DS89C430, DS89C440, and DS89C450 devices. The timer simulation was previously based on a older data book and therefore incorrect.
- **[Target Debugging for Analog Devices ADuC83x and ADuC84x]**  
**Options - Project - Debug - Use: ADI Monitor Driver** selects target driver for the Analog Devices ADuC834 and ADuC84x devices. This driver directly connects via a serial COM interface to the on-chip Download/Debug Kernel of the ADuC device. No specific monitor or firmware is required. A detailed documentation is available in the **Analog Devices ADuC83x/84x Download/Debug Driver User's Guide** (..\C51\HLP\MonADI.CHM).
- **[Support for Infineon XC866]**  
Added complete support for new Infineon XC800 Device series including device simulation, ULINK driver, and MCBXC866 Evaluation Board support.
  - Example projects are provided in the folder **..\C51\Examples\Infineon XC866**.
  - Complete documentation is available in the **MCBXC866 User's Guide** (..\C51\HLP\MCBXC866.CHM) which also explains the ULINK driver.
- **[Support for ST uPSD34xx Series]**  
Added complete ULINK support for new ST uPSD34xx device series.
- **[C Library]**
  - Corrected: on Dallas 390, 400, 5240, and 5250 devices **tan (INF)** generated a wrong return, and **printf ("%f")** did not print the floating point value INF and NaN.
  - Corrected: the return value of scanf and sscanf was just 0xFF instead of -1 when no arguments where processed.
  - Corrected: the library takes care about the changed behavior of the Memory Accelerator on **Dallas DS80C390 Rev. C**.
  - Corrected: on Dallas 390, 400, 5240, and 5250 devices memmove in xdata memory failed when source buffer is overlapping destination buffer.

## C51 Version 7.50a Release

- **[LX51 Linker/Locater]**
  - Corrected: potential problem with Linker Code Packing that might cause in-efficient operation or a MEMORY SPACE OVERLAY warning.
  - Corrected: fixup error messages when using the **REMOVEUNUSED** directive.
  - Corrected: when using L51\_BANK.A51 the far memory addressing used a wrong offset (-64KB). Instead of X:0x20000 the X:0x10000 was addressed.
- **[BL51 Linker/Locater]**  
The new BL51 Linker/Locater was by accident not included in Version 7.50.
- **[C Library]**  
Corrected a problem on Dallas 390, 400, 5240, and 5250 devices with **log** and **log10** when const data is not in stored in segment 0 (C:0x0000-C:0xFFFF).

## C51 Version 7.50 Release

### ▪ [uVision2 Debugger]

Added Extended Memory Simulation for Mentor M8051EW. Refer to **Application Note 171: Using M8051EW Memory Extension** in the folder **\C51\EXAMPLES\M8051EW**.

Added peripheral simulation support and target debugging dialogs for the following devices:

- [Dallas Semiconductor DS89C420](#),
- [Dallas Semiconductor DS89C430](#),
- [Dallas Semiconductor DS89C440](#),
- [Dallas Semiconductor DS89C450](#),
- [Silicon Labs C8051F000](#),
- [Silicon Labs C8051F001](#),
- [Silicon Labs C8051F002](#),
- [Silicon Labs C8051F005](#),
- [Silicon Labs C8051F006](#),
- [Silicon Labs C8051F007](#),
- [Silicon Labs C8051F010](#),
- [Silicon Labs C8051F011](#),
- [Silicon Labs C8051F012](#),
- [Silicon Labs C8051F015](#),
- [Silicon Labs C8051F016](#),
- [Silicon Labs C8051F017](#),
- [Silicon Labs C8051F350](#),
- [Silicon Labs C8051F351](#),
- [Silicon Labs C8051F352](#),
- [Silicon Labs C8051F353](#),
- [SST SST89E554RC](#),
- [SST SST89E564RD](#),
- [SST SST89V554RC](#),
- [SST SST89V564RD](#).

### ▪ [ISD51 In-System Debugger]

Added example configuration for Dallas DS89C420, DS89C430, DS89C440, and DS89C450 devices. For details refer to [Application Note 177: Using ULINK with STMicroelectronics Turbo uPSD 3300/3400 Devices](#) and the sample projects in the folder **..\C51\EXAMPLES\ST uPSD**.

### ▪ [C51 Compiler]

- Corrected a code generation issue for negative array index values. For example:

```
signed int i = v1 - v2;  
value = table [i + 4]; // code works for 'signed char' index
```

```
// but fails with 'signed int' index
```

- Added **MODC2 Compiler Directive**: enables use of dual data pointers available on the Cast and Evatronix R80515 core. Using additional data pointers improves the performance of the following library functions: **memcpy**, **memmove**, **memcpy**, **strcpy**, and **strcmp**.
- Added **MODH2 Compiler Directive**: enables use of dual data pointers available on Hynix, ST uPSD 33xx, and ST uPSD 34xx devices. Using additional data pointers improves the performance of the following library functions: **memcpy**, **memmove**, **memcpy**, **strcpy**, and **strcmp**.
- **[CX51 Compiler]**  
Corrected the following problems for the SmartMX instruction set:
  - Switch/case with **long** types and **ROM(HUGE)**.
  - Over optimization with **CMPPW** instruction.
  - Stack adjustment failure with **setjmp/longjmp** library routines.
- **[C Library]**  
Corrected several library problems including:
  - Corrected a problem on Dallas 390, 400, 5240, and 5250 devices with **asin**, **acos**, and **atan** when const data is not in stored in segment 0 (C:0x0000-C:0xFFFF).
  - The **labs** function has been optimized and is now fully reentrant.
  - Added configuration symbol ?C?DPSEL that defines DPSEL SFR address for **MOD517(NOAU)** multiple DPTR support. May be used for Mentor M8051EW based devices which have the Infineon method for multiple DPTR but with a different DPSEL SFR address. The ?C?DPSEL definition shown below may be included in a chip-specific Startup.A51 file. Without this definition, the DPSEL register is accessed at the default address 0x92.

```
PUBLIC ?C?DPSEL
?C?DPSEL DATA 0A2H ; define DPSEL address for Mentor M8051EW
```

- **[LX51 Linker]**  
Added the **REMOVEUNUSED** (abbreviation **RU**) directive which removes unused program and data segments provided that Data Overlaying is enabled.
- **[ULINK]**  
Added instruction trace support to the STMicroelectronics uPSD ULINK Driver. For details refer to [Application Note 177: Using ULINK with STMicroelectronics Turbo uPSD 3300/3400 Devices](#) and the sample projects in the folder **..\C51\EXAMPLES\ST uPSD**.
- **[BL51/LX51 Linker]**  
Improved the **OVERLAY** directive. Now, using **OVERLAY (\* ! (func1, func2))**, you may combine the segments of several function call trees. This is useful for interrupt functions that have overlayable data but use the same interrupt level. Such interrupt functions cannot interrupt each other. Therefore, data overlaying of both call trees is possible. For example:

```
void irq0 (void) interrupt 0 {
    unsigned char arr[10];
    arr[0] = 0;
}

void irq1 (void) interrupt 1 {
    unsigned char arr[10];
    arr[0] = 0;
}
```

If irq0 and irq1 are set to the same priority level their data areas may be overlaid. The **OVERLAY** directive may be specified to do that as follows:

```
BL51 ... OVERLAY (* ! (irq0, irq1))
```

The linker map file shows the following OVERLAY MAP.

SEGMENT	DATA_GROUP	
+-> CALLED SEGMENT	START	LENGTH
-----		
?PR?IRQ1?I	0008H	000AH
*** NEW ROOT *****		
?PR?IRQ0?I	0008H	000AH

Note that both functions' call trees are overlaid.

## C51 Version 7.20 Release

### ▪ [uVision2 Debugger]

Added peripheral simulation support for the following devices:

- [Analog Devices ADuC841](#),
- [Analog Devices ADuC842](#),
- [Analog Devices ADuC843](#),
- [Analog Devices ADuC845](#),
- [Analog Devices ADuC847](#),
- [Analog Devices ADuC848](#),
- [Altium Nexar TSK51x](#) MCU Core,
- [Cast/Evatronix R8051](#) MCU Core,
- [Cast/Evatronix R80515](#) MCU core,
- [DCD DR8051](#) MCU core,
- [DCD DR80390](#) MCU core,
- [DCD DR8051XP](#) MCU core,
- [DCD DR80390XP](#) MCU core,
- [Mentor M8051EW](#) MCU core,
- [TI MSC1200Y2](#),
- [TI MSC1200Y3](#).

CPU core simulation may be expanded with user peripherals using the [AGSI Interface](#).

### ▪ [ISD51 Configuration Examples]

Added ISD51 configuration for the TI MSC1200. Refer to [Application Note 183: Use ISD51 on TI MSC1200](#) for more information.

### ▪ [ISD51 Configuration Examples]

Revised the configuration for Philips LPC900 to match the current Philips LPC900 device versions.

### ▪ [FlashMonitor]

Added a new configuration for the [Atmel AT8xC5122](#). Current configurations support almost all new Atmel device variants and are summarized in the **\KEIL\C51\FashMon\ReadMe.txt** file.

- **[C51 Compiler]**  
Added intrinsic functions **\_push\_** and **\_pop\_** that may be used to save and restore **sfr** registers in interrupt functions.
- **[C51 Compiler]**  
Added support for the [Silicon Labs C8051F12x](#) multiply and accumulate unit (MAC0). The **MDU\_F120** directive enables use of MAC0 for int and long multiplication and long shift operations.
- **[CX51 Compiler]**  
Corrected a problem with far pointers and long arithmetic.
- **[ULINK Driver for ST µPSD series]**  
Corrected a problem while programming Flash memory that caused the microcontroller to execute random instructions. The MCU is now forced into RESET while programming the Flash.

## C51 Version 7.10 Release

- **[uVision2 Debugger]**  
Added simulation support for the following devices:
  - Philips P89LPC935 and other downgraded LPC900 devices
  - Philips P89C669
  - ST µPSD33xx
- **[uVision2 Debugger]**  
Corrected A/D converter simulation of the ADuC831.
- **[LX51 Linker Code Packing]**  
Corrected a problem with code packing and JMP optimizations.
- **[ULINK Driver for ST µPSD series]**  
Merged the Flash and Debug Setup Dialog. The ST Merge Utility (UTLADRM.EXE) may be called automatically. Additionally, a problem with lock-ups was corrected.
- **[C51 Compiler]**  
Corrected a problem with unbalanced PUSH/POP sequences in complex indirect function calls.

## C51 Version 7.09 Release

- **[uVision2 Debugger - ULINK Support for STMicroelectronics µPSD3300/3400]**  
Added final release of the ULINK debugging and Flash programming support for the new **STMicroelectronics µPSD3300/3400** devices. Program examples and an [Application Note](#) are provided in the **\KEIL\C51\EXAMPLES\ST uPSD** folder.
- **[LX51 Linker Code Packing]**  
Corrected a potential problem with linker code packing which may cause the linker/locator to hang.

## C51 Version 7.08 Release

- **[C51 Compiler]**  
Corrected a potential problem with the CSTXPTR function of the EEPROM program examples for the Atmel parts (`\KEIL\C51\EXAMPLES\FarMemory\E2PROM on T89C51RD2` and `\KEIL\C51\EXAMPLES\FarMemory\3 XData Areas on T89C51RD2`).
- **[Cx51 Compiler]**  
When the **setjmp** and **longjmp** library routines are used in a code banking application, you must include the source file `\KEIL\C51\LIB\SETJMP.A51` in your project. This file contains versions of these routines that support code banking. The routines in the standard library do not support code banking applications.
- **[Cx51 Compiler]**  
Corrected a problem passing complex function parameters when far variables are used as function arguments.
- **[A51/AX51 Macro Assembler]**  
Modified the assembler so that core SFR register symbols (ACC, B, DPL, DPH, PSW, SP) are automatically defined even when the **NOMOD51** directive is used. This avoids error messages when generating assembler source (SRC) files from C modules that do not include a register definition file.
- **[AX51 Macro Assembler]**  
Added the **EVEN** directive which is described in the Ax51 User's Guide. Previously, this directive was available only in the A251 Assembler.
- **[LX51 Linker]**  
Corrected a problem with the memory allocation strategy that was introduced in Version 7.07. This problem caused incorrect address calculations for constant segments that were located after packed code segments.
- **[LX51 Linker]**  
Corrected a problem with wildcards in the **SEGMENTS** directive. For example, **SEGMENTS (?PR?\*?module (C:0x4000))** did not locate all segments above 0x4000. Instead only the first segment was located at C:0x4000 and other segments were located within the **CLASS** definition.
- **[ISD51 In-System Debugger]**  
Corrected a problem with flash breakpoints when the flash block size (CBLK\_SZ) was configured for 1 byte.
- **[Cx51 Run-Time Library]**  
Improved the **rand** library routine to deliver better distributed pseudo-random numbers. The new algorithm is based on a galois LFSR generator.
- **[uVision2 Debugger/Simulator]**  
Added simulation for the following IP Cores:
  - [Actel Core8051](#).
  - [Cast C8051 Core](#), [Cast D80530 Core](#), [Cast R8051 Core](#), and [Cast R80515 Core](#).
  - [Dolphin Flip8051 Breeze](#), [Dolphin Flip8051 Cyclone](#), [Dolphin Flip8051 Thunder](#), and [Dolphin Flip8051 Wind](#).
- **[uVision2 Debugger]**  
Added ULINK debugging and Flash programming support for the new **STMicroelectronics μPSD3300/3400** series of devices. Also added Flash programming support for the μPSD3200 series of devices. Program examples and an [Application Note](#) are provided in the `\KEIL\C51\EXAMPLES\ST uPSD` folder.
- **[uVision2 IDE]**  
Corrected a problem with menu and shortcut configuration on Windows NT machines.

- **[uVision2 Debugger/Simulator]**  
Corrected simulation and target display problems with the Philips LPC900 series (i.e. the DIVM factor).

## C51 Version 7.07 Release

- **[uVision2 IDE]**  
Added a new dialog for project component management under **Project - Components, Environment and Books**. This dialog allows you to change the order of project targets and file groups.
- **[uVision2 Debugger/Simulator]**  
Added simulation for the following devices:
  - Atmel AT89C5131 (except USB)
  - Atmel AT89C5132 (except A/D Converter, USB, Audio Interface, MMC Controller, and IDE/ATAPI Interface)
  - Cygnal C8051F300
  - Cygnal C8051F301
  - Cygnal C8051F302
  - Cygnal C8051F303
  - Cygnal C8051F304
  - Cygnal C8051F305
  - Cygnal C8051F310
  - Cygnal C8051F311
  - Cygnal C8051F320
  - Cygnal C8051F321
  - Cygnal C8051F330
  - Cygnal C8051F331
  - TI MSC1210 (I<sup>2</sup>C Simulation)
  - TI MSC1211
  - TI MSC1212
- **[uVision2 Debugger/Simulator]**  
Corrected the following simulation problems:
  - Analog Devices ADuC831: The internal RC clock (32768 Hz) is used for the watchdog timer. This was incorrectly documented in the first datasheets.
  - Dallas Semiconductor Devices: The Watchdog EWT reset is now performed only on power-up reset and not on every reset.
  - Winbond Devices: The Watchdog EWT reset is now performed only on power-up reset and not on every reset.
- **[uVision2 Debugger]**  
Corrected problems viewing local variables in projects linked with the LX51 Extended Linker/Locater.
- **[C51 Compiler]**  
Corrected problems in Dallas Contiguous Mode with `*(ptr++)` type constructs and far pointer initialization at file level.

- **[CX51 Compiler]**  
Corrected problems for Philips MX when far/generic pointer assignments are reused in the subsequent statements. For example:  
  

```
unsigned char far * far pbMemory;
unsigned long volatile ulAddress;

pbMemory += 1L;           // uses ?C?PSTPTR to save the result
ulAddress = pbMemory;     // uses wrong result of the previous ?C?PSTPTR library function
```
- **[C51 Compiler]**  
Corrected problems with SRC output on extended 8051 platforms like Dallas Contiguous Mode and Philips 51MX.
- **[C51 Compiler]**  
Added the ability to locate **far** memory variables in HDATA memory using the **\_at\_** keyword.
- **[LX51 Linker]**  
Improved memory allocation in linker code packing to reduce the size of segment gaps when the **SEGMENTS** directive is used with assembler segments.
- **[Ax51 Macro Assembler]**  
Added the **DEFINE** command line directive. This directive allows you to supply C preprocessor on the command line. The syntax is identical to that of the Cx51 Compiler.
- **[Monitor-390]**  
Corrected problems with xdata memory updates and added support for Dallas 400 and Dallas 5240 devices.
- **[ISD51 In-System Debugger - Version 2]**  
Corrected a problem that caused serial break to fail when configured for non-Flash breakpoints. Added example for TI MSC121x devices in the **\KEIL\C51\EXAMPLES\TI MSC121X** folder.

## C51 Version 7.06 Release

- **[ISD51 In-System Debugger - Version 2]**  
Corrected a problem that caused serial break to fail on ISD51 when configured for non-Flash breakpoints.
- **[Flash Monitor-51 Version 4]**  
Added configurations for Atmel AT89C51RD2 and AT89C51SND1.
- **[Flash Monitor-51 Version 4]**  
Corrected a problem that caused Break on Serial Interrupt to fail when the monitor was generated using older versions of the tools.
- **[uVision2 Debugger/Simulator]**  
Added simulation for the following devices:
  - Atmel AT89C51RD2
  - Atmel AT89C51ED2
  - Atmel AT89C51ID2
  - Dallas Semiconductor 80C530
  - Philips 8xC652



- Philips 8xC654
- **[uVision2 Debugger/Simulator]**  
Corrected simulation problems with the CCU Timer on the Philips P89LPC932.
- **[uVision2 Debugger/Simulator]**  
Added context menu commands in the Source Window and Disassembly Window for Set Program Counter, Show Disassembly, and Show Source Code.
- **[Cx51 Compiler]**  
Corrected a problem that caused **far** pointer comparisons to NULL to fail when the statement immediately following re-used the same pointer.
- **[Cx51 Compiler]**  
Corrected a problem with the **NOINTPROMOTE** directive that caused the compiler to generate incorrect code. This problem was introduced in C51 V7.04 in an effort to correct another integer promotion problem.

## C51 Version 7.05 Release

- **[ISD51 In-System Debugger - Version 2]**  
Added several new features including:
  - Real-Time Flash Breakpoints using In-System Application Programming (IAP),
  - User I/O via serial debugging interface,
  - Address range support in the memory verify function.
- **[Flash Monitor-51 - Version 4]**  
A new variant of Monitor-51 is included in the PK51 Professional Developers Kit. The new Monitor runs on unmodified Flash Devices that provide IAP programming. It requires no von-Neumann memory and can run from the on-chip resources of standard 8051 Flash Devices. The Flash Monitor includes Flash download and real-time breakpoint support. Currently the Monitor is pre-configured for the Atmel T89C51CC01, T89C51RC2, and T89C51RD2 but it can easily be configured for other devices.
- **[uVision2 Debugger/Simulator]**  
Added simulation support for the Atmel T89C51CC02, T8xC5115, AT89C1051, AT89C1051U, AT89C2051, and AT89C4051 devices.
- **[uVision2 Debugger/Simulator]**  
Added simulation support for the Philips P8xC51Rx2, P8xC51RB2H/RC2H/RD2H, and P8xC3xX2 devices.
- **[uVision2 Debugger/Simulator]**  
Corrected a simulation problem with the AT89S8252 EEPROM and Dual DPTR access.
- **[uVision2 Debugger/Simulator]**  
Corrected a simulation problem with the ADuC832 ADC DMA Stop.
- **[uVision2 Debugger]**  
Corrected a problem with local variables not displaying in the watch window - locals tab.
- **[LX51 Extended Linker]**  
Corrected a potential problem with linker code packing on optimize level 10 and 11.
- **[C51 Compiler]**

Corrected erroneous combining of identical end sequences of while (1) loops.

## C51 Version 7.04 Release

- **[Philips MX Device Support]**  
Changed INIT\_MX.A51. Now, initialization is enabled for **far variables** by default. **far variables** may now be absolutely located with **\_at\_**.
- **[MON390 Monitor for Dallas Contiguous Mode]**  
Corrected problems with breakpoints above 64KB code. Corrected a problem with single-stepping in switch/case statements.
- **[uVision2 Target Debugger]**  
Added call stack display and step out command for classic 8051 devices. Added [EPM900](#) Emulator/Programmer support for Philips 89LPC9xx Devices.
- **[C51 Compiler]**  
Corrected integer promotion problems with combined pointer and char arithmetic. For example:  

```
int xdata *Test (int xdata * adr, unsigned char a, unsigned char b)  {
    return adr+(a+b);          // did not promote 'a+b' to int
}
```
- **[A51 Assembler, AX51 Assembler]**  
Added support for C-style bitwise operators (| (OR), & (AND), ~ (NOT)) to the A51 Assembler and AX51 assembler. These operators are useful for common C and assembler header files that use **#define** statements.

## C51 Version 7.03a Release

- **[RTX51 Tiny2]**  
Corrected os\_wait problems on K\_IVL, K\_TMO+K\_SIG events. Refer to **KEIL\C51\RTX51TINY2\README.TXT** for details.
- **[uVision2 Debugger/Simulator]**  
Added **Cygnal 80C51F02x** device simulation.

## C51 Version 7.03 Release

- **[C51 Compiler]**  
Corrected incorrect warnings on enum mismatches.
- **[LX51 Linker]**  
Corrected fixup error messages on Dallas 390 target.

## C51 Version 7.02 Release

- **[LX51 Linker]**  
Corrected problems with linker code packing and code banking and a potential problem with the bank switch table location in banking mode 4.

- **[LX51 Linker]**  
Added the **SPEEDOVL** directive to makes LX51 and BL51 compatible for data overlaying. Detailed information on **SPEEDOVL** is available in the Assembler/Utilities User's Guide, Chapter 9, Control Summary.
- **[BL51 Linker, LX51 Linker, Libraries]**  
Added support for Mentor M8051EW Memory Extension that provides access to 1MB ROM and 1MB RAM. The **IBANKING** directive supports the on-chip banking hardware on M8051EW-based devices and is available in both the BL51 Linker and the LX51 Linker. Additionally, LX51 may be configured with the L51IBANK.A51 file that supports a 64KB bank for constants and 16 x 64KB banks for **far** variables. Refer to the **\C51\EXAMPLES\M8051EW** folder for example code and additional information.
- **[C51 Compiler, CX51 Compiler]**  
Corrected a potential problem with generic and far pointer comparisons to a NULL pointer constant.
- **[C51 Compiler, CX51 Compiler]**  
Enhanced warning messages for **enum** and **memory-typed pointer** assignments.
- **[Monitor for Dallas Contiguous Mode]**  
Released MON390 which provides a target monitor for the Dallas Contiguous Mode. Detailed information, pre-configured Monitor versions, and example programs may be found in the **\C51\MON390** folder.
- **[uVision2 Debugger/Simulator]**  
Added simulation for several new devices (Atmel 89C51Ix and the Cygnal 80C51F02x). The Cygnal 80C51F02x devices are currently in beta status.
- **[uVision2 IDE]**  
Added Flash menu to uVision2. This menu provides a direct interface to external Flash programming tools like Philips FlashMagic. Flash programming commands are configured under **Options for Target Utilities**.
- **[RTX51 Tiny Version 2]**  
Released RTX51 Tiny Version 2. This release contains several new enhancements like code banking support and cooperative task switching.

## C51 Version 7.01 Release

- **[C51 Compiler]**  
Added support for Extended Call Return Mode (ECRM) available in the new Philips 51MX devices. This mode is configured in **START\_MX.A51**. It enhances the code density of the **ROM(HUGE)** memory model. This optimization requires that [Linker Code Packing](#) is enabled. Once enabled, ACALL, LCALL, and ECALL instructions are optimized.
- **[C51 Compiler]**  
Corrected minor problems in the **ROM(HUGE)** memory model.
- **[uVision2 IDE]**  
Added several new devices to the uVision2 device database.
- **[uVision2 IDE]**  
Added peripheral simulation for the new Philips 89LPC932.
- **[uVision2 IDE]**  
Added peripheral simulation for the second UART in Winbond devices.

- **[uVision2 IDE]**  
Added peripheral simulation for the four priority levels in the new version of the Philips 8xC552 device.
- **[C51 Compiler]**  
Corrected problems in **L51\_BANK.A51** with regards to variable code banking on classic 8051 devices that used standard banking hardware.
- **[C51 Compiler]**  
Corrected a syntax problem (that was introduced in Version 7.00) in the **setjmp.h** header file.
- **[LX51 Linker]**  
Validated LX51 [Linker Code Packing](#) for Philips 51MX and Dallas 390/400 devices.
- **[BETA RELEASE]**  
Released RTX51 Tiny Version 2 BETA with the following new features and enhancements:
  - Code Banking Support
  - Explicit Task Switch Function
  - RUN Status Flag
  - CPU IDLE Mode Support
  - Hooks for Adding User Code to the RTX51 Tiny Hardware Timer Interrupt
  - Improved Handling for Interval Events
  - Reduced Code and Data Size
  - Improved Performance
- **[BETA RELEASE]**  
Released MON390 BETA which provides a target monitor for the Dallas Contiguous Mode. Detailed information, pre-configured Monitor versions, and example programs may be found in the **\C51\MON390** folder.

## C51 Version 7.00 Release

- **[C51 Compiler]**  
Added the **ROM(HUGE)** directive which provides support for the Philips 51MX Linear Programming Model. Select this option in uVision2 using Options for Target-Code Rom Size: Huge: 8MB program. More information is available in [Application Note 160](#): Programming the Philips 51MX Architecture with the Keil PK51.
- **[C51 Compiler]**  
Added the ability to perform 24-bit arithmetic calculations using **far** pointers. For more information, refer to [Application Note 160](#): Programming the Philips 51MX Architecture with the Keil PK51.
- **[ISD51]**  
Released [ISD51 \(In-System Debugger\)](#): a new target debugger that may be linked to user applications. Refer to **\C51\ISD51** for more information.
- **[LX51 Linker]**  
Added a new LX51 Linker-Level Optimization called [Linker Code Packing](#). This optimization analyzes and reduces total program size. In uVision2, enable this optimization in Options for Target - C51: Code Optimization: Linker Code Packing. This optimization is available for all projects even those that use code banking. Note that this optimization is still a BETA RELEASE for the Philips 51MX and Dallas 390/400 devices.

- **[C51 Compiler]**  
Added two new optimizations to the C51 Compiler that reduce program code size. In uVision2, enable these optimization in Options for Target - C51: Code Optimization: Level.
- **[Lx51 Linker]**  
Added Linker [Disassembly Output File](#). This output file contains the complete disassembly of your application complete with intermixed high-level source and all addresses. In uVision2, enable this option in Options for Target - Listing - Linker Code Listing.

## C51 Version 6.23 Release

- **[C51 Compiler]**  
Corrected problems with register optimizations in while loops.
- **[C51 Compiler]**  
Corrected problems implicitly casting types in ternary statements.
- **[C51 Compiler]**  
Enhanced performance of the run-time library and pointer operations for the Dallas Semiconductor 80C390 Contiguous Mode.
- **[C51 Compiler]**  
Added ability to locate XDATA and CODE in regions other than 00:xxxx for the Dallas Semiconductor 80C390.
- **[C51 Compiler]**  
Corrected various problems with initializations with Lx51 and bit objects.
- **[A51 Assembler]**  
Corrected problems [synchronizing MPL Macros](#) while debugging.
- **[OHx51 Object File Converter]**  
Added the **MERGE32K** directive which generates merged HEX files for hardware with 32K common areas. Refer to the Ax51 User's Guide, Chapter 9, Bank Switch Configuration for more information. In uVision2, select this option in Options for Target - Output - Merge32K HEX File.
- **[OHx51 Object File Converter]**  
Corrected problems with Lx51 HEX file generation.
- **[uVision2 IDE]**  
Added simulation support for Analog Devices MicroConverters.
- **[BETA RELEASE]**  
[ISD51 In-System Debugger](#) is a new target debugger that may be linked to user applications. Refer to **\C51\ISD51** for more information.
- **[BETA RELEASE]**  
Added a new LX51 Linker-Level Optimization called [Linker Code Packing](#). This optimization analyzes and reduces total program size. In uVision2, enable this optimization in Options for Target - C51: Code Optimization: Linker Code Packing.
- **[BETA RELEASE]**  
Added two new optimizations to the C51 Compiler that reduce program code size. In uVision2, enable these optimization in Options for Target - C51: Code Optimization: Level.

- **[BETA RELEASE]**  
Added Linker [Disassembly Output File](#). This output file contains the complete disassembly of your application complete with intermixed high-level source and all addresses. In uVision2, enable this option in Options for Target - Listing - Linker Code Listing.

## C51 Version 6.22 Release

- **[C51 Compiler]**  
Corrected several problems that were introduced with Dynamic Register Allocation in Version 6.21.
- **[C51 Compiler]**  
Added new examples for the const far memory type.
- **[C51 Compiler]**  
Added support for the extended stack in the Analog Devices MicroConverters.
- **[Lx51 Linker]**  
Added examples for **far const** memory with classic 8051 devices in **\C51\EXAMPLES\FARMEMORY\1MB CONSTANTS ON CLASSIC 8051**. These examples show how to use memory banking with text constants.

## C51 Version 6.21 Release

- **[C51 Compiler]**  
Corrected several minor problems.
- **[C51 Compiler]**  
Added the **MODDA** directive and library support for the Dallas 390 Math Accelerator.
- **[C51 Compiler]**  
Added [dynamic register allocation](#) optimization which reduces program size and data usage.
- **[C51 Compiler]**  
Added [switch/if path combination](#) optimization.
- **[C51 Compiler]**  
Added optimization for [long 0 comparisons](#).
- **[C51 Compiler]**  
Corrected several optimizer problems that were introduced in C51 V6.20.
- **[BL51 Linker]**  
Corrected several minor problems.
- **[uVision2 IDE]**  
Added simulation support for the Dallas Semiconductor 80C390 peripherals.

## C51 Version 6.20 Release

- **[C51 Compiler]**  
Enhanced the **L51\_BANK.A51** file to support even larger code banking programs (up to 4MB).
- **[C51 Compiler]**  
Added enhanced optimization for register variables.
- **[C51 Compiler]**  
Added variable banking support for classic 8051 devices.
- **[uVision2 IDE]**  
Added debug dialogs for classic 8051 devices.
- **[uVision2 IDE]**  
Added four 64KB user memory areas (S:, T:, U:, and V:) that may be used when debugging.
- **[uVision2 IDE]**  
Added functions or for special simulation capabilities (E<sup>2</sup>PROM, I<sup>2</sup>C communication, and so on).
- **[uVision2 IDE]**  
Improved the Version Control (SVCS) Connection and corrected several problems with environment variables.
- **[uVision2 IDE]**  
Added several new items to the Help Menu.
- **[uVision2 IDE]**  
Added several project management enhancements.
- **[uVision2 IDE]**  
Added numerous chips to the Device Database.
- **[uVision2 IDE]**  
Added simulation support for the on-chip peripherals of the following devices (complete information is available in the  $\mu$ Vision2 Help Menu - Simulated Peripherals item.):
  - Infineon C504.
  - Infineon C505C.
  - Infineon C508.
  - Infineon C515A.
- **[uVision2 IDE]**  
Added new debugging dialogs for MON51.

## C51 Version 6.14 Release

- **[C51 Compiler]**  
Added examples to demonstrate the Dallas 390 contiguous mode (\C51\EXAMPLES\DALLAS 390).
- **[C51 Compiler]**

Added examples to demonstrate the Philips 51MX architecture 16MB support (\C51\EXAMPLES\PHILIPS 80C51MX).

- **[C51 Compiler]**  
Added **far** memory (above 64K) support using the \C51\LIB\XBANKING.A51 configuration file.
- **[C51 Compiler]**  
Added **far** memory examples for the 16MB memory space of the Analog Devices ADuC812 and for the AtmelWMM 89C51RD E2PROM area (\C51\EXAMPLES\FARMEMORY).
- **[C51 Compiler]**  
Added macros for absolute **far** memory access in **ABSACC.H**.
- **[Cx51 Compiler]**  
Added the **INCDIR** directive where you may specify include paths.
- **[Ax51 Assembler]**  
Added the **INCDIR** directive where you may specify include paths.
- **[uVision2 IDE]**  
Added simulation support for the on-chip peripherals of the following devices:
  - Infineon C509.
  - Infineon C517A.
  - Infineon C515C.

## C51 Version 6.12 Release

- **[C51 Compiler]**  
Removed the 256-symbol limit from OMF51 object files.
- **[C51 Compiler]**  
Extended the length of C variable names to 256 characters.
- **[uVision2 IDE]**  
Added simulation support for the on-chip peripherals of the following devices:
  - Atmel WM T87C5111.
  - Atmel WM T87C5112.

## C51 Version 6.11 Release

- **[C51 Compiler]**  
Added the **ROM(512K)** and **ROM(D16M)** directives to select the contiguous modes of the Dallas Semiconductor 80C390. This directive must be used with the **OMF2** directive. Refer to \C51\EXAMPLES\DALLAS390\README.TXT for more information.

## C51 Version 6.10 Release



- **[C51 Compiler]**  
Finalized support for the Philips 80C51MX.
- **[C51 Compiler]**  
Finalized support for the Dallas Semiconductor 80C390.
- **[C51 Compiler]**  
Added banking mode 4 to **L51\_BANK.A51** for user-provided bank switching macros.

## C51 Version 6.03 Release

- **[BETA RELEASE]**  
Added support for the Philips 80C51MX, Dallas Semiconductor 80C390, and LX51 Extended Linker.
- **[C51 Compiler]**  
Added the **OMF2** directive which causes the C51 Compiler to output object files for the LX51 Extended Linker. This new object file format supports 16MB code space for constants and 16MB xdata space and is required for extended 8051 device variants like the Analog Devices ADuC812, Dallas 390, and others.
- **[BL51 Linker]**  
Added a new warning if your program CODE or XDATA exceeds the specified memory area size.
- **[uVision2 IDE]**  
Added simulation support for the on-chip peripherals of the following devices:
  - Dallas 320/323/520/530.
  - Philips 80C552/554.
  - Temic 89C51RD2 (including on-chip E<sup>2</sup>PROM).
  - Temic 80C51CC02.
- **[uVision2 IDE]**  
Integrated debugger interface for the Triscend E5 CSoC. Refer to **\C51\HLP\README\_FOR\_TE5\_UV2.TXT** for more information.

## C51 Version 6.02 Release

- **[C51 Compiler]**  
Corrected several minor problems with Optimizer Level 9.
- **[uVision2 IDE]**  
Added simulation support for the on-chip peripherals of the following devices:
  - Analog Devices ADuC812.
  - Philips LPC Devices.
- **[uVision2 IDE]**  
Added simulation support for multiple DPTR registers.

## C51 Version 6.01 Release

- **[C51 Compiler]**  
Added the **RET\_ISTK**, **RET\_PSTK**, and **RET\_XSTK** directives which unload the on-chip stack and use the reentrant stack for storing return addresses.

Refer to [The RET\\_ISTK Directive](#), [The RET\\_PSTK Directive](#), or [The RET\\_XSTK Directive](#) for more information.

- **[C51 Compiler]**  
Added ANSI library routines `modf`, `strtod`, `strtol`, and `strtoul`. Refer to the on-line compiler manual for more information.

- **[BL51 Linker]**  
Modified the **CODE** and **XDATA** directives to allow address ranges. For example:

```
BL51 my_prog.obj CODE(0x0000-0x3FFF, 0x8000-0xFFFF)
```

- **[BL51 Linker]**  
Modified segment control directives to allow wildcards in segment names. For example:

```
BL51 my_file.obj (CODE (?PR?*?my_file (0x100))
```

Locates all program code segments in the module `my_file` to address 0x100 and up.

- **[uVision2 IDE]**  
Added debugger support for [MON51](#).
- **[uVision2 IDE]**  
Added simulation support for almost all 40-pin DIP devices (8051FC, RD, RD+, 8052, and so on).
- **[uVision2 IDE]**  
Added support for syntax coloring in assembler code.
- **[uVision2 IDE]**  
Added item in the context menu to insert the CPU register definition file.
- **[uVision2 IDE]**  
Added context-sensitive help for library routines and error messages. To get help for a library routine, position the cursor on the function and press the **F1** key. To get help for an error or warning message, select the message and press the **F1** key.

## C51 Version 6.00 Release

- **[C51 Compiler]**  
Added 3 new optimizer levels which shrink program size up to 15%.
  - Optimize Level 7 (Extended Access Optimization) uses DPTR for register variables. Pointer and array accesses have been optimized for both speed and code size.
  - Optimize Level 8 (Reuse of Common Entry Code) moves common function entry code to the beginning of a function which saves code memory. Optimize (8) is the new default optimization level for C51 Version 6.xx.
  - Optimize Level 9 (Common Block Subroutines) detects and packs multiple-instruction sequences into subroutines. This optimization is most efficient on

large source files.

- **[C51 Compiler]**  
Added specific header file support for almost all devices.
- **[C51 Compiler]**  
Added configuration file (`\C51\LIB\CONF151.A51`) for the Intel 151.
- **[C51 Compiler]**  
Updated the enum type to automatically adjust to 8 or 16 bits.
- **[C51 Compiler]**  
Added dual data pointer support for Atmel devices (AT89S8252). Use the **MODA2** directive to enable dual data pointer support. Use the **NOMODA2** directive to disable support.
- **[C51 Compiler]**  
Added dual data pointer support for Philips devices. Use the **MODP2** directive to enable dual data pointer support. Use the **NOMODP2** directive to disable support.
- **[C51 Compiler]**  
Added dual data pointer support for Temic devices. Use the **MODP2** directive to enable dual data pointer support. Use the **NOMODP2** directive to disable support.
- **[C51 Compiler]**  
C51 now ensures that register bank 0 is selected for interrupts declared without the **using** attribute. The instruction `MOV PSW, #0` is added to these routines.

Previously, you were required to add the **using 0** attribute to high-priority interrupts when low-priority interrupts used a different register bank. This was the case for RTX51 Full and RTX51 Tiny applications.

If your application uses only register bank 0, you may use the **ONEREBANK** directive to specify that the C51 compiler does not generate the additional `MOV PSW, #0` instruction.

- **[A51 Assembler]**  
Added C preprocessor support that expands text before the source file is assembled. Directives like **#if**, **#else**, **#endif**, and **#include** are supported in assembly source code (refer to the C51 User's Guide, Chapter 4). The **#include** file path is obtained from the **C51INC** environment variable.
- **[A51 Assembler]**  
Added the following predefined Macros:
  - **\_\_FILE\_\_**: Name of the file being assembled.
  - **\_\_LINE\_\_**: Current line number in the file being assembled.
  - **\_\_TIME\_\_**: Time when the assembly was started.
  - **\_\_DATE\_\_**: Date when the assembly was started.
  - **\_\_STDC\_\_**: Defined to 1.
  - **\_\_A51\_\_**: Version number of the A51 Assembler (for example 600 for V6.00).
  - **\_\_KEIL\_\_**: Defined to 1.

- **[A51 Assembler]**

Added support for C-style sfr and sbit declarations. The A51 Assembler now accepts standard C-style register definition files. This allows you to use the same header files for your C and assembler source files. The following sfr and sbit declarations were added:

```
sfr P0 = 0x80;
sbit P0_1 = P0^1;
```

- **[A51 Assembler]**

Added error output using the **\_\_ERROR\_\_** directive. For example:

```
IF CVAR1LEN > 10
    __ERROR__ "CVAR1 LEN EXCEEDS 10 BYTES"
ENDIF
```

Or using the C-style preprocessor.

```
#ifdef TESTVERS && RELEASE
#error TESTVERS GENERATED IN RELEASE MODE
#endif
```

- **[A51 Assembler]**

Added the **INCDIR** (abbreviation **ID**) directive where you may specify the paths to assembler include files. You may specify one or more paths to search when a **\$INCLUDE** directive is processed. The search order for **\$INCLUDE** is:

- Current directory (typically, the folder of the uVision2 project file).
- Paths specified with **\$INCDIR**.
- Path derived from the bin folder using **..\ASM (\C51\ASM)**.

For example:

```
$INCDIR (C:\C51\ASM)
A51 STARTUP.A51 INCDIR (C:\C51\INC,C:\MYDIR)
```

The search order for **#include** is identical to that used by the C51 Compiler.

- **[BL51 Linker]**

Added the **DISABLEWARNING** directive (abbreviation **DW**) which allows you to selectively disable linker warning messages. For example:

```
BL51 myfile.obj DISABLEWARNING (1,5)
```

Disables warnings 1 and 5.

- **[BL51 Linker]**

BL51 now sorts and locates segments according to their length. This ensures fewer gaps in memory. Use the **NOSORTSIZE** directive (abbreviation **NOSO**) to disable this feature.

- **[BL51 Linker]**

Added the **SPEEDOVL** directive (abbreviation **SP**) which causes the linker to ignore references to constant segments that start with ?CO?. This speeds up the overlay process but may result in a lack of warnings with regard to constant segments. This could lead to problems if you use pointers to functions and do not manually specify the call tree references to the linker. Refer to [SPEEDOVL Directive](#) and [Application Note 129: Function Pointers in C51](#) for more details.

This directive may be useful for applications with complex pointer to function tables.

▪ **[BL51 Linker]**

Added the **RECURSIONS** directive (abbreviation **RC**) which allows you to specify the maximum number of recursive calls allowed before the linker aborts. The default number of recursions allowed is 10.

A recursive call generates the following linker warning:

```
*** WARNING L13: RECURSIVE CALL TO SEGMENT
```

When the maximum number of recursions is exceeded, the linker responds with the following error:

```
FATAL ERROR 232: APPLICATION CONTAINS TOO MANY RECURSIONS.
```

To use this directive, enter the following on the linker command line or in the Misc box in uVision2.

```
BL51 test.obj RECURSIONS (100)
```

Note that the linker may run a long time to detect all recursions and remove the references. Unless you have specific reasons to change this setting, you should leave it at the default level of 10.

▪ **[BL51 Linker - Code Banking]**

Added the **NOAJMP** directive (abbreviation **NOAJ**) which disables use of the AJMP instruction in the inter-bank jump table in code banking programs. This option is required for 8051 derivatives which do not support the AJMP instruction.

▪ **[BL51 Linker - Code Banking]**

Added the **NOINDIRECTCALL** directive (abbreviation **NOIC**) which specifies that function pointers do not access functions outside the current code bank (in code banking programs). By default, in code banking programs the BL51 Linker inserts inter-bank calls into the call table for functions called through a function pointer. If your application uses function pointers and if you can ensure that indirect calls never cross a code bank you may use the **NOINDIRECTCALL** directive to save space in the call table.

Refer to [The Code Banking Mechanism](#) for more information about the scheme used by the BL51 Linker for code banking.

▪ **[BL51 Linker - Code Banking]**

Added the **NOJMPTAB** directive (abbreviation **NOJT**) which specifies that the call tree is not created for code banking applications. This feature is provided for developers who will create their own code banking scheme. This directive modifies the behavior of the BL51 Linker as follows:

- The L51\_BANK.A51 code banking logic file is not required.
- The jump and call instructions are not modified for banked functions.
- No warnings are generated if a jump or call is made to another code bank.

If you use this directive you must ensure that the proper bank is selected before a call is performed. The BL51 Linker no longer selects the code bank.

## Example Programs

Example programs included in the **\C51\EXAMPLES** folder demonstrate how to use the uVision3 Project Manager and Debugger (see the uVision3 Quick Start Guide for details). Please refer to these if you are new to the tools and want to get started quickly.

## Device Database™

A unique feature of the Keil uVision3 IDE is the [Device Database™](#) which contains information about more than 1000 supported microcontrollers. When you create a new uVision3 project and select the target chip from the database, uVision2 sets all assembler, compiler, linker, and debugger options for you. The only option you must configure is the memory map.

As new devices become available, they are added to the database along with data sheets and header files. For information about adding your own chips to the database or about creating your own personal databases refer to the following knowledgebase articles.

- [Adding Custom Parts to the Device Database](#)
- [Creating Custom Databases](#)

## Peripheral Simulation

The uVision4 Debugger provides [complete simulation](#) for the CPU and on-chip peripherals of most embedded devices. To discover which peripherals of a device are supported, in uVision4 select the Simulated Peripherals item from the Help menu. You may also use the web-based [Device Database™](#). We are constantly adding new devices and simulation support for on-chip peripherals so be sure to check Device Database™ often.

## Technical Support

At Keil Software, we are dedicated to providing you with the best development tools and technical support. That's why we offer numerous ways you can get the technical support you need to complete your embedded projects.

- [Technical Support Knowledgebase](#)  
More than 2500 technical support questions and answers are available in the Support Solutions Knowledgebase. When a new question arises, it is added to the knowledgebase which is continuously published to the Web. This enables you to get technical support at times when our support staff is unavailable.
- [Application Notes](#)  
Numerous Application Notes help you decipher complex features and implement robust applications.
- [Example Programs and Files](#)  
Utility programs, example code, and sample projects are regularly added to the Download File section of the web site.
- [Discussion Forum](#)  
Post questions, comments, and suggestions to the Keil Software Discussion Forum and interact with other Keil users around the world.

Many of the features of our Technical Support Knowledgebase and Web Site are the results of your suggestions. If you have any ideas that will improve them, please [give us your feedback!](#)

## Contact Details

If you experience any problems or have any questions about this product, contact one of our [distributors](#) or offices for assistance.

**In the USA...****ARM, Inc.**

4965 Preston Park Blvd, Suite 650  
Plano, TX 75093  
USA

800-348-8051 - Sales  
972-312-1107 - Support  
972-312-1159 - Fax

[sales.us@keil.com](mailto:sales.us@keil.com) - Sales E-Mail  
[support.us@keil.com](mailto:support.us@keil.com) - Support E-mail

**In Europe...****ARM Germany GmbH**

Bretonischer Ring 16  
D-85630 Grasbrunn  
Germany

+49 89 456040-0 - Sales  
+49 89 456040-24 - Support  
+49 89 468162 - Fax

[sales.intl@keil.com](mailto:sales.intl@keil.com) - Sales E-Mail  
[support.intl@keil.com](mailto:support.intl@keil.com) - Support E-Mail

---

Copyright © 2012 ARM Ltd and ARM Germany GmbH.  
All rights reserved.  
Visit our web site at [www.keil.com](http://www.keil.com).