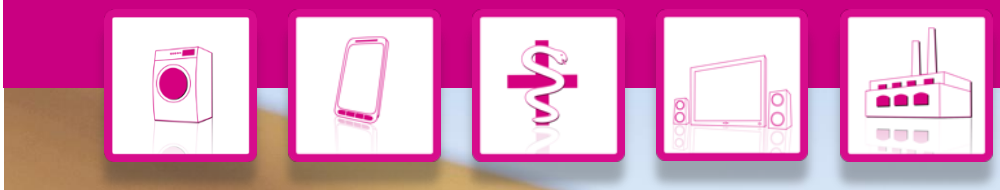
The Analog Mixed Signal Company

Technical Note

Rev. 1.00 / August 2014

ZWIR4512

Migration Guide for Updating to ZWIR4512 Library V1.9



Automotive ASICs and Industrial ASSPs

Interface ICs

Multi-Functional and Robust

Contents

1 Introduction 3
2 Release Notes 3
3 Migrating Existing Projects 4
3.1. Update Linker Script 4
3.2. Remove the Startup Code File 7
3.3. Update Project Setting 8
3.4. Update NetMA 10
4 Related Documents 11
5 Glossary 11
6 Document Revision History 12

List of Figures

Figure 3.1 Updated Linker Script Required by New Stack 4
Figure 3.2 Removing Startup Code from the Project 7
Figure 3.3 Switching GCC Target to arm-unkown-eabi 8
Figure 3.4 Update System Include Directories 9
Figure 3.5 Adding NetMA1 10

1 Introduction

This application note provides a migration guide for updating the network stack libraries for the ZWIR4512 Secure Low-Power Wireless IPv6 Module, including the new NetMA2 library. The network stack libraries have been updated to be compatible with the latest versions of Rowley Associate's CrossStudio integrated development environment and the GNU Compiler Collection (GCC). This update also requires changes in the customer project setup. This guide describes the steps needed to modify the user's project to support the new version 1.9 library.

Advantages of the new library:

- Support for more development environments with only one library.
- Simplified project setup and structure—less complexity
- Smart new methods for maintaining and administering large networks

2 Release Notes

	Changes for ZWIR4512 Library Upgrade
1.	EABI: The ZWIR4512 stack is now compiled with *EABIv5*.
2.	The enums are now short enums.
3.	The NetMA implementation from previous releases has been moved to a separate library: *libZWIR45xx-NetMA1.a.*
4.	A new NetMA version with additional features has been added: *libZWIR45xx-NetMA2.a*.
5.	**`ZWIR_OTAU_ErrorCode_t`**, **`ZWIRSEC_eDroppedICMP`** and **`ZWIRSEC_eDropedPacket`** indicate if an ICMP or other packet was dropped by an IPsec rule. **`ZWIRSEC_eUnknownSPI`** indicates that an IPsec packet was received but no associated security association was found. With active replay check, **`ZWIRSEC_eReplayedPacket`** indicates a replayed packet. IPsec indicates authentication vector mismatches (corrupted packet) with **`ZWIRSEC_eCorruptedPacket`**.
6.	The startup code is now part of the stack.
7.	The linker script has been updated with a storage section for NetMA2 and application data.
8.	Support for external real time clock (RTC) oscillator has been added.
9.	The RTC is now providing a one-second tick interval.
10.	The first backup register is used by *libZWIR45xx-6LoWPAN.a* for storing RTC status information and must not be used by the application code.
11.	For **`ZWIR_PowerDown()`**, the sleep time is a multiple of one second for all power down modes.
12.	There is a new low power 8-MHz mode for **`ZWIR_MCUFrequency_t`** (**`ZWIR_mcu8MHzLowPower`** = 4), which is used after startup. This mode provides better performance and lower power consumption via an optimized flash controller setup and a disabled PLL.
13.	All deprecated functions and enums are removed from *ZWIR45xx-IPsec.h.*
14.	**`ZWIRSEC_AddSecurityAssociation()`** requires an additional parameter **`uint8_t replayCheck`** to enable or disable replay checks for security associations.
15.	Replay check is enabled automatically for IKEv2 negotiated security associations.

	Changes for ZWIR4512 Library Upgrade
16.	There are new stack functions: ▪ **ZWIR_SelectRTCSource()** ▪ **ZWIR_GetRTC()** ▪ **ZWIR_SetRTC()** ▪ **ZWIR_GetParameter()**
17.	New **ZWIR_SystemParameter_t**: ▪ **ZWIR_spNeighborRetransTime** ▪ **ZWIR_spDoAddressAutoConfiguration**

3 Migrating Existing Projects

To use the new library it is necessary to change all existing ZWIR projects. For a clean setup, it is recommended that a new empty project be created and all old project source files be copied and added to the new project.

However, it is also possible to update existing projects by performing the following steps.

3.1. Update Linker Script

The new stack requires an updated linker script. The new linker script provides more meaningful names, support for the new NetMA2 protocol, and improved memory size checks.

To update an existing project, the project linker script (*ZWIR4512.ld*) must be replaced with the linker script shown in Figure 3.1 (file is included in the release files). If the project linker script contains project specific modifications, the modifications must be added to the new linker script.

Figure 3.1 Updated Linker Script Required by New Stack

```
1       /*****************************************************************************/
2       /*  Linker command file for STM32                                            */
3       /*****************************************************************************/
4
5       /* Memory Definitions STM32F103RC  */
6       MEMORY {
7         FLASH (xr)      : ORIGIN = 0x08000000, LENGTH = 0x00040000
8         stack (xw)      : ORIGIN = 0x20000000, LENGTH = 0x00001400
9         ram   (xw)      : ORIGIN = 0x20001400, LENGTH = 0x0000AC00
10      }
11
12      /* GCC LD bugfix */
13      __originStack = ORIGIN ( stack ) + 0;
14      __originFlash = ORIGIN ( FLASH ) + 0;
15      __originRam = ORIGIN ( ram ) + 0;
16
17      /** Number of flash pages reserved for non-volatile parameter storage
18       *  This memory will not be affected by firmware over the air updates
19       */
20      __nvReservedPageCount = 1;
21
22      /* Define size of one page of flash memory*/
23      __mcuFlashPageSize = 0x800;
24
25      /* Define stack start address*/
26      __estack = __originStack + LENGTH ( stack );
27
28      /* Compute the length of flash required for non-volatile parameter storage */
29      __nvReservedFlashSize = __nvReservedPageCount * __mcuFlashPageSize;
30
```

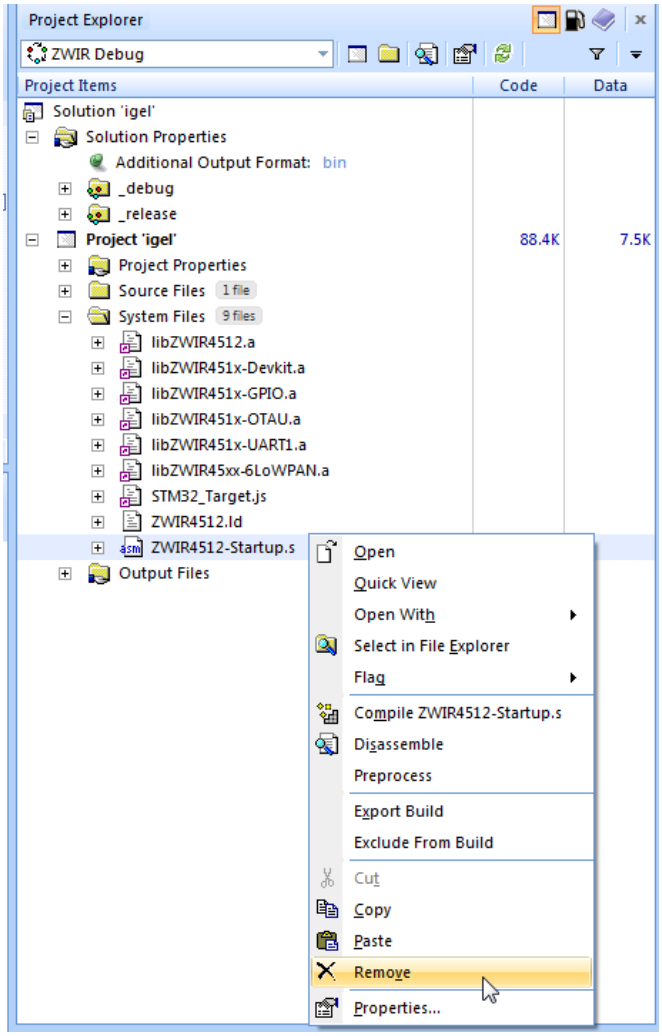
```
31      SECTIONS {
32
33        .update_code : {
34          _otau_start = .;
35          KEEP ( *(.otau_isr_vectors .otau_isr_vectors.*) );
36          start = .;
37          KEEP ( *(.otau_startup .otau_startup.*) );
38          KEEP ( *(.otau_permanent .otau_permanent.*) );
39        } >FLASH
40        __otauUsed = SIZEOF (.update_code) > 0;
41
42        ASSERT ( SIZEOF (.update_code) < 0x7c0, "Invariant OTAU code too big (maximum allowed length is 0x7c0 bytes)" )
43
44        .interface_seg :  {
45          . = __otauUsed ? 0x7c0 - SIZEOF (.update_code) : 0;
46          KEEP ( *(.otau_interface .otau_interface.*) );
47        } >FLASH
48
49        .status_seg ALIGN ( __mcuFlashPageSize ):  {
50          KEEP ( *(.otau_status_seg .otau_status_seg.*) );
51        } >FLASH
52
53        /* vector-table containing initial SP and PC */
54        .isr_vector ALIGN ( __mcuFlashPageSize ) : {
55          __ldISRVectorStart__ = .;
56          KEEP ( *(.isr_vectors) )
57        } >FLASH
58
59        .boot_init : {
60          start = (start != _otau_start) ? start : . ;
61          *(.boot_init .boot_init.*)
62        } >FLASH
63
64        .init : {
65          . = ALIGN ( 4 );
66          *(.init .init.*)
67          . = ALIGN ( 4 );
68        } > FLASH
69
70        .text : {
71          . = ALIGN ( 4 );
72          *(.text .text.* .glue_7t .glue_7 .gnu.linkonce.t.* .gcc_except_table)
73          *(.rodata)
74          *(.rodata*)
75          *(.rodata.str1.4)
76          . = ALIGN ( 4 );
77        } > FLASH
78
79        .ARM.extab ALIGN ( 4 )  : { *(.ARM.extab* .gnu.linkonce.armextab.*) } > FLASH
80         __exidx_start = .;
81        .ARM.exidx ALIGN ( 4 )  : { *(.ARM.exidx* .gnu.linkonce.armexidx.*) } > FLASH
82         __exidx_end = .;
83        _end = .;
84
85        __RAM_startup_init_data__ = ALIGN ( 4 );
86
87      /*****************************************************************************
88      *                                    DATA
89      *****************************************************************************/
90         .otau_var(NOLOAD) : {
91           *(.otau_var.ZMDI_otauFirmwareSeg1)
92           *(.otau_var.ZMDI_otauPageCache)
93           *(.otau_var.ZMDI_otauFirmwareSeg2)
94           *(.otau_var);
95         } >ram
96
97        .data : AT ( __RAM_startup_init_data__ ) {
98          __RAM_data_start__  = .;
99          *(.data .data.* .gnu.linkonce.d.*)
100         *(.init_array.00000)
101         *(.init_array)
102         . = ALIGN(4);
103         __RAM_data_end__ = .;
104       } > ram
105
106       .tdata : AT ( __RAM_startup_init_data__ + SIZEOF(.data)) {
107         __TRAM_data_start__  = .;
108         *(.tdata .tdata.*)
109         . = ALIGN(4);
110         __TRAM_data_end__ = .;
111       } > ram
112
113       __RAM_all_data_end__ = .;
```

```
114
115       _DATASize = __RAM_all_data_end__ - __RAM_data_start__;
116       /*****************************************************************************
117     *                                    .bss
118     *****************************************************************************/
119       .bss (NOLOAD) : {
120         __bss_start__ = .;
121         *(.bss .bss.*)
122         __bss_end__ = . ;
123       } >ram
124
125       .tbss (NOLOAD) : {
126         __tbss_start__ = .;
127         *(.tbss .tbss.*)
128         __tbss_end__ = . ;
129       } >ram
130
131       PROVIDE ( end = . );
132       PROVIDE ( __heap_start__ = . );
133       PROVIDE ( __heap_end__ = __originRam + LENGTH ( ram ) );
134
135       /* This is to circumvent an gcc 4.4.x issue */
136       /DISCARD/ : { *(.eh_*) }
137
138     /*****************************************************************************
139     *                                    .store
140     *****************************************************************************/
141       /* set location counter to EndOfFlash - NonVolatileMemorySize */
142       . = __originFlash + LENGTH(FLASH) - __nvReservedFlashSize;
143
144       /* Do not rename .nvDataMemory */
145       .nvDataMemory ALIGN ( __mcuFlashPageSize ) : {
146         *(.NetMA_NVParameters .NetMA_NVParameters.*)
147
148         /*************************************************************************
149         * Uncomment the following section to use a nonvolatile store section.
150         * Don't forget to update __nvReservedPageCount accordingly.
151         *************************************************************************
152           . = NEXT ( __mcuFlashPageSize );
153           __store_start__ = .;
154                *(.store .store.*)
155                __store_end__ = .;
156         *************************************************************************/
157
158       } > FLASH
159     }
160
161     __mcuFlashEnd = __originFlash + LENGTH(FLASH);
162
163     __executableSize = __RAM_startup_init_data__ + _DATASize - __originFlash;
164
165     /* Executable segment - this contains the current firmware code */
166     __executableSegmentStart = __ldISRVectorStart__;
167
168     /* This variable defines the maximum firmware code size */
169     __totalFirmwareSpace = LENGTH ( FLASH ) - __nvReservedFlashSize - ( __executableSegmentStart - __originFlash );
170     __maxExecutableSize = __otauUsed ? __totalFirmwareSpace / 2 : __totalFirmwareSpace;
171
172     /* Compute the maximum number of pages a firmware image may occupy:
173       ( flash size - invariant OTAU code - non-volatile data memory ) / 2 */
174     __otauMaximumImagePageCount = __maxExecutableSize / __mcuFlashPageSize;
175
176     /* Update segment - this contains a copy of the new firmware */
177     __updateSegmentStart = __executableSegmentStart + ( __otauMaximumImagePageCount * __mcuFlashPageSize );
178
179     /*check if there is enough space in flash memory*/
180     ASSERT( ( __executableSize < __maxExecutableSize ), "error: binary size too big!");
181
182     /* set thumb bit for entry symbol*/
183     start = start | 1;
184
185     ENTRY ( start );
186
187     /* Force the linker to link library symbols which would otherwise not be included*/
188     EXTERN ( g_pfnVectors ZMDI_OTAU_Interface NetMA_RXHandler NetMA_ParameterStore);
```

3.2. Remove the Startup Code File

Since the startup code is now part of the library, the *ZWIR4512-Startup.s* must be removed from all projects. This can be done by right clicking on the *ZWIR4512-Startup.s* file inside the Project Explorer and then select “Remove” from the drop-down menu.

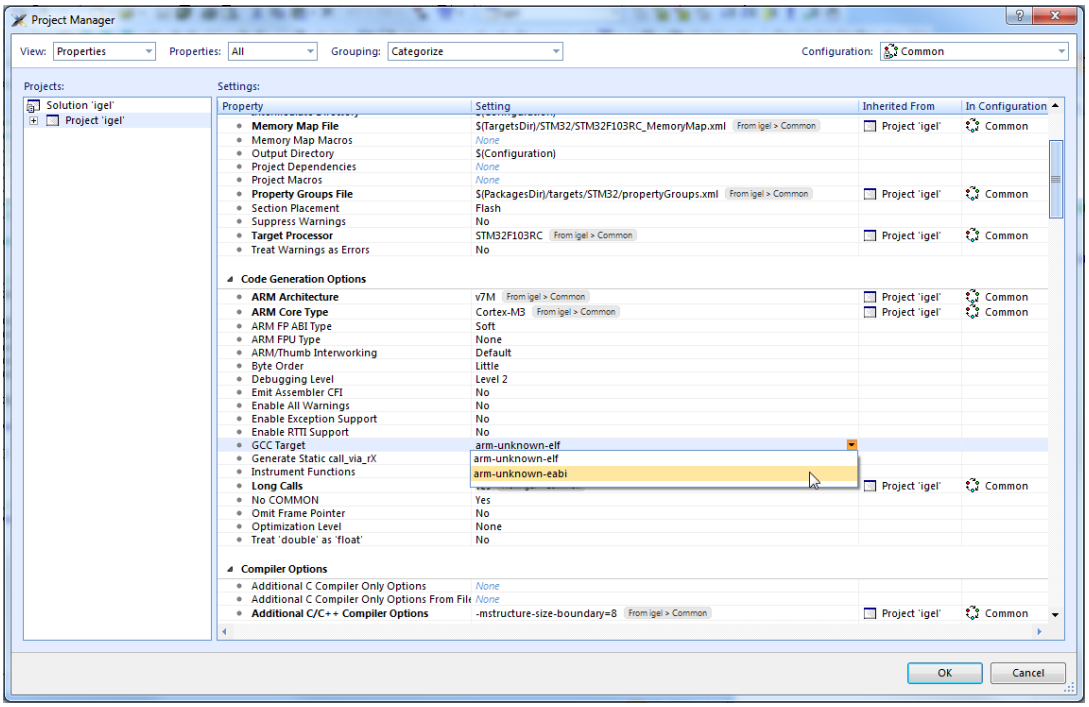
Figure 3.2 Removing Startup Code from the Project



3.3. Update Project Setting

Because the ZWIR4512 library is compiled with *EABIv5*, the project setting “GCC Target” must be changed to “arm-unknown-eabi.”

***Figure 3.3 Switching GCC Target to* arm-unkown-eabi**

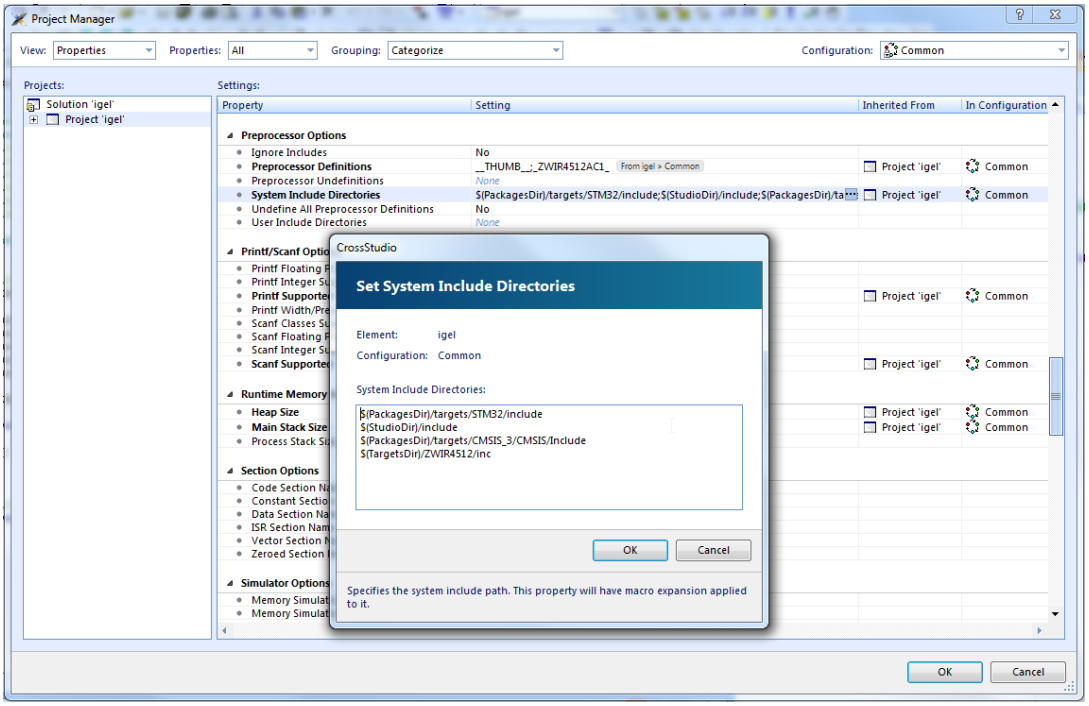


The “System Include Directories” setting must be updated as well.

The new directories are

```
$(PackagesDir)/targets/STM32/include
$(StudioDir)/include
$(PackagesDir)/targets/CMSIS_3/CMSIS/Include
$(TargetsDir)/ZWIR4512/inc
```

Figure 3.4 Update System Include Directories



3.4. Update NetMA

The previous NetMA1 functionality has been moved to a separate library:

```
libZWIR45xx-NetMA1.a
```

If an existing project uses the current NetMA implementation, this library must be added to the project. The easiest way to add the library is to open the project `.hzp` file with a text editor and add the following line in the systems file section for each project:

```
<file file_name="$(TargetsDir)/ZWIR4512/Release/libZWIR45xx-NetMA1.a">
  <configuration Name="Common" file_type="Library"/>
</file>
```

After updating the file, the project must be reopened and rebuilt.

Figure 3.5 Adding NetMA1

```
    <folder Name="System Files" file_name="">
      <file file_name="$(TargetsDir)/STM32/STM32_Target.js">
        <configuration Name="Common" file_type="Reset Script"/>
      </file>
      <file file_name="system/ZWIR4512.ld">
        <configuration Name="Common" file_type="Linker Script"/>
      </file>
      <file file_name="$(TargetsDir)/ZWIR4512/Release/libZWIR4512.a">
        <configuration Name="Common" file_type="Library"/>
      </file>
      <file file_name="$(TargetsDir)/ZWIR4512/Release/libZWIR45xx-6LoWPAN.a">
        <configuration Name="Common" file_type="Library"/>
      </file>
      <file file_name="$(TargetsDir)/ZWIR4512/Release/libZWIR451x-UART1.a">
        <configuration Name="Common" file_type="Library"/>
      </file>
      <file file_name="$(TargetsDir)/ZWIR4512/Release/libZWIR451x-OTAU.a">
        <configuration Name="Common" file_type="Library"/>
      </file>
      <file file_name="$(TargetsDir)/ZWIR4512/Release/libZWIR45xx-NetMA1.a">
        <configuration Name="Common" file_type="Library"/>
      </file>
    </folder>
  </project>
```

4 Related Documents

Note: *X.x.pdf* refers to the latest version of the document.

Document	File Name
ZWIR4512 Data Sheet	*ZWIR4512 Data Sheet_revX.x.pdf*
*ZWIR45xx Serial Command Interface User Manual**	*ZWIR45xx_SCI_Manual_revX.x.pdf*
*ZWIR451x Programming Guide**	*ZWIR451x_ProgGuide_revX.x.pdf*
*ZWIR45xx Application Note – Enabling Firmware Over the Air Updates**	*ZWIR45xx_AN_OTAU_Update_revX.x.pdf*
*ZWIR45xx Application Note – Using IPSec and IKEv2 in 6LoWPANS**	*ZWIR45xx_AN_Security_revX.x.pdf*

Visit the ZWIR4512 product page www.zmdi.com/zwir4512 on ZMDI's website www.zmdi.com or contact your nearest sales office for the latest version of these documents.

* Documents marked with an asterisk require a free customer login account. To set up a login account, click on Login in the upper right corner of the web page and follow the instructions.

5 Glossary

Term	Description
6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks
EABI	Embedded Application Binary Interface
GCC	GNU Compiler Collection
ICMP	Internet Control Message Protocol
IKEv2	Internet Key Exchange version 2
IPSec	Internet Protocol Security
PLL	Phase-Locked Loop
RTC	Real-Time Clock
NetMA	Network Monitoring and Administration

6 Document Revision History

Revision	Date	Description
1.00	August 24, 2014	First release of document.

Sales and Further Information	www.zmdi.com	wpan@zmdi.com		
Zentrum Mikroelektronik Dresden AG Global Headquarters Grenzstrasse 28 01109 Dresden, Germany Central Office: Phone +49.351.8822.306 Fax +49.351.8822.337	**ZMD America, Inc.** 1525 McCarthy Blvd., #212 Milpitas, CA 95035-7453 USA USA Phone 1.855.275.9634 Phone +1.408.883.6310 Fax +1.408.883.6358	**Zentrum Mikroelektronik Dresden AG, Japan Office** 2nd Floor, Shinbashi Tokyu Bldg. 4-21-3, Shinbashi, Minato-ku Tokyo, 105-0004 Japan Phone +81.3.6895.7410 Fax +81.3.6895.7301	**ZMD FAR EAST, Ltd.** 3F, No. 51, Sec. 2, Keelung Road 11052 Taipei Taiwan Phone +886.2.2377.8189 Fax +886.2.2377.8199	**Zentrum Mikroelektronik Dresden AG, Korea Office** U-space 1 Building 11th Floor, Unit JA-1102 670 Sampyeong-dong Bundang-gu, Seongnam-si Gyeonggi-do, 463-400 Korea Phone +82.31.950.7679 Fax +82.504.841.3026
European Technical Support Phone +49.351.8822.7.772 Fax +49.351.8822.87.772 **European Sales** (Stuttgart) Phone +49.711.674517.55 Fax +49.711.674517.87955				