Instruction

Z-Wave Programmer User Guide

Document No.:	INS10679
Version:	10
Description:	This document describes how to use the Zensys USB Z-Wave Programmer for non-volatile memory programming of the Z-Wave 100/200/300/400 Series Single Chips. The Programmer supports also programming of the external EEPROM on the Z-Wave modules.
Written By:	JFR;SSE;DDA;ANI
Date:	2011-09-27
Reviewed By:	JKA;EFH;CHL;SDH
Restrictions:	Partners Only

Approved by:

Date	CET	Initials	Name	Justification
2011-09-27	11:39:14	NTJ	Niels Thybo Johansen	

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
1	20061001	SSE JFR	ALL	Initial draft
2	20070322	JFR	Section 5.8.7 Section 5.10.1	Shortcut keys revised Console frequency parameters revised
3	20070420	SSE	Section 5.11	Programmer firmware upgrade added
4	20080227	JFR	Section 5.3.6	Revised with respect to 100 Series
5	20081212	DDA JFR	ALL	Revised and updated with respect to 400 Series and ZDP03A
6	20090929	JFR	Section 3.1.3 Section 3.1.3 & 4.4.2	ZDP03A Jumper J20 settings clarified Not possible to program external EEPROM on 100 Series based modules when using ZDP03A
7	20100127	DDA	Section 4.5 Section 5.7 Section 5.8 Section 4.5, 5.5, 5.8 Sections 4.5.2, 5.4 Section 5.10 Section 5.11 Section 5.10.2 Section 4.5.1 Section 5.12	Updated screenshot, added 4.5.3 MTP description added Added Hold device in reset, Write on PCB button, Use test frequencies. Updated screenshots. Description of SRAM updated, screenshot updated Console interface syntax added. Firmware upgrade updated. Updated Console interface syntax Updated section Section added
8	20110210	DDA	Section 5.3.5	Section about Calibration and Program feature added.
9	20110217	DDA	Section 5.10	Console interface syntax updated.
10	20110816	DDA	Section 5.3.6 Section 5.12	Added ZW040x RF settings Updated interface

Table of Contents

CONFIDENTIAL

Table of Figures

Table of Tables

1 ABBREVIATIONS

Abbreviation	Explanation
EEPROM	Electrically Erasable Programmable Read Only Memory
Flash	A type of constantly-powered nonvolatile memory that can be erased and reprogrammed in units of memory called blocks. It is a variation of electrically erasable programmable read-only memory (EEPROM) which, unlike flash memory, is erased and rewritten at the byte level, which is slower than flash memory updating.
GUI	Graphical User Interface
HW	Hardware
ISP	In-System Programmer
OTP	A One-Time Programmable memory
PA	Power Amplifier
pF	Pico Farad
RF	Radio Frequency
Rx	Receive
SW	Software
Tx	Transmit
USB	Universal Serial Bus
ZM1206	6 cm² Z-Wave Module containing a 100 Series Z-Wave Single Chip
ZM1220	20 cm² Z-Wave Module containing integrated PCB antenna, EEPROM and 100 Series Z-Wave Single Chip
ZM2102	2 cm² Z-Wave Module containing a 200 Series Z-Wave Single Chip
ZM2106	6 cm² Z-Wave Module containing a 200 Series Z-Wave Single Chip
ZM2120C	20 cm² Converter Z-Wave Module, used to convert ZM2106 to ZM2120-formfactor
ZM3102N	2 cm² Z-Wave Module containing a 300 Series Z-Wave Single Chip
ZM3106C	6 cm² Z-Wave Module containing a 300 Series Z-Wave Single Chip
ZM3120C	20 cm² Converter Z-Wave Module, used to convert ZM3106C to ZM3120-formfactor
ZM4225	25 cm² Z-Wave Module containing integrated PCB antenna, EEPROM and Z-Wave ZW0402 Single Chip (QFN)
ZW0102	100 Series Z-Wave Single Chip
ZW0201	200 Series Z-Wave Single Chip
ZW0301	300 Series Z-Wave Single Chip
ZW040x	400 Series Z-Wave Single Chips
MTP	Many Time writeable Memory

2 INTRODUCTION

2.1 Purpose

The purpose of this document is to give a description of the Zensys USB Z-Wave Programmer for non-volatile memory programming of the Z-Wave 100/200/300/400 Series Single Chips. The Z-Wave Programmer also supports programming of the external EEPROM on the Z-Wave modules. Finally, the Z-Wave Programmer can also be used to configure transmission power and RF settings on the Z-Wave modules and Lock Bits.

2.2 Audience and prerequisites

The audience is Z-Wave partners and Zensys.

3 GETTING STARTED

3.1 Installation

The Z-Wave Programmer can be used for programming the non-volatile memory in the Z-Wave 100/200/300/400 Series Single Chips during SW and HW development and for small production series. The Z-Wave Programmer can be used for writing data such as home ID to the external EEPROM on the Z-Wave modules. The Z-Wave Programmer can also be used to configure transmission power and RF settings on the Z-Wave modules.

The Z-Wave module can be mounted in the Z-Wave module socket on the Z-Wave Programmer or via an ISP cable to a ZW0x0x Interface Module hosting the Z-Wave module. The Z-Wave Programmer is connected to the PC via the USB interface and is powered by an external power supply or USB interface depending on type of Z-Wave Programmer and jumper settings.

3.1.1 System Requirements

Minimum PC requirements to run Z-Wave Programmer software are as follows:

- Windows XP, Service Pack 2
- Microsoft .NET Framework 2.0 or later
- USB interface

3.1.2 Hardware installation using ZDP02A Z-Wave Programmer

The ZDP02A Z-Wave Programmer can be used for programming the flash in the Z-Wave 100/200/300 Series Single Chips. The ZDP02A Z-Wave Programmer is shown on the figure below:

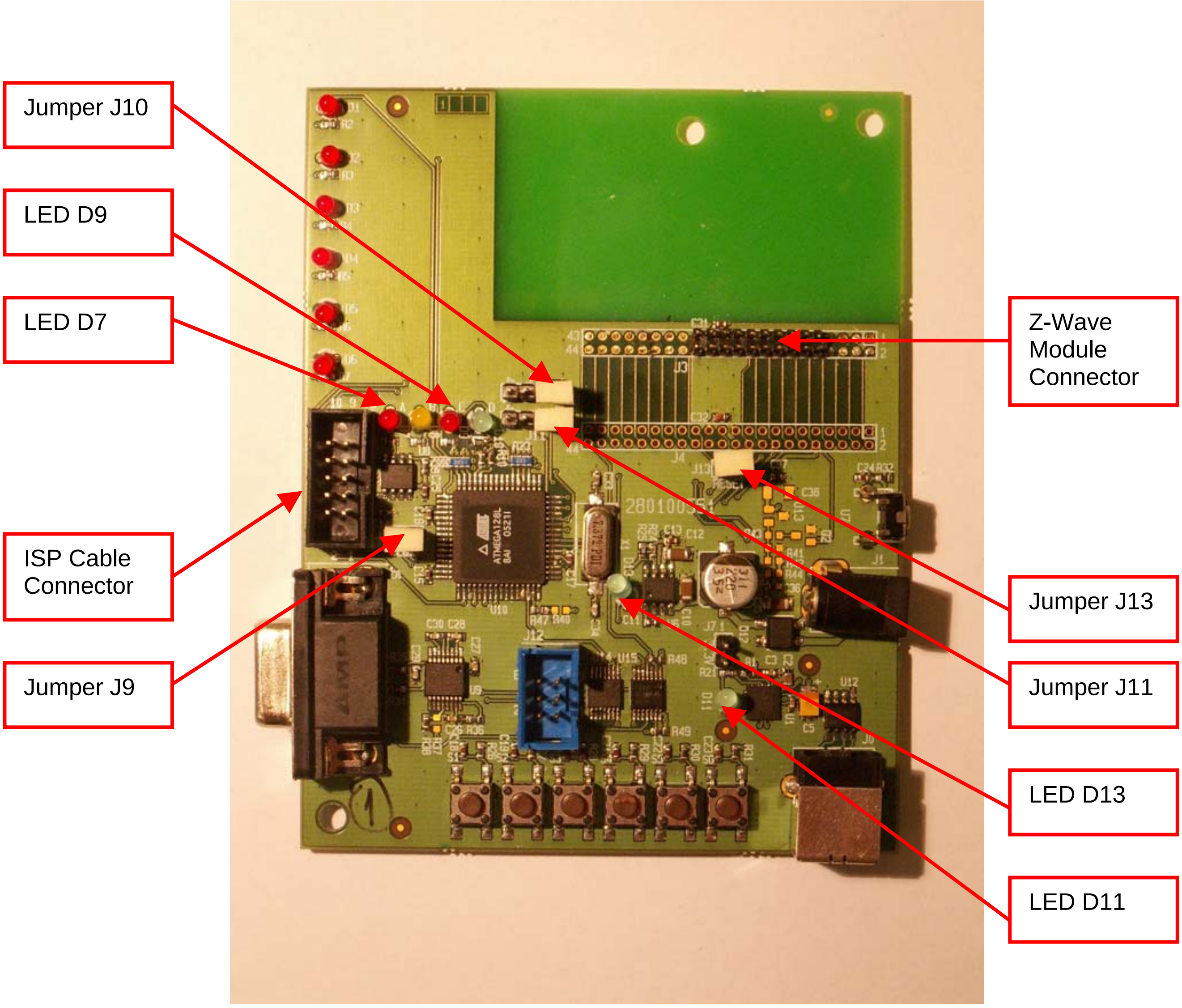


Figure 1. The ZDP02A Z-Wave Programmer connectors, LEDs and jumpers layout.

Jumpers J10, J11 and J13 must be shorted as shown on Figure 1.

Jumper J9 must be shorted in case the Z-Wave module connected via the ISP cable is powered from the Z-Wave programmer. Jumper J9 must be open in case the Z-Wave module connected via the ISP cable uses an independent power-supply.

IMPORTANT: Check carefully that the jumpers are mounted correctly on the Z-Wave Programmer HW.

Several LED’s are used to indicate the state of the programmer. LED11 is on if USB controller power is on and there is a connection to the host PC. LED D13 is on if the board is powered on. LED7 is on when the Z-Wave Programmer firmware is running. LED D9 is on when the Z-Wave Programmer is busy and it is off when the programmer is idle.

The figure below shows the ZDP02A Z-Wave Programmer HW with a Z-Wave module mounted in the Z-Wave Module connector. The Z-Wave Programmer is connected to the PC using the USB interface and powered by an external power supply. The ZDP02A Z-Wave Programmer can not be powered via the USB interface, but the ZDP03A can. The module can be mounted in the Z-Wave Module connector while debugging the embedded application.

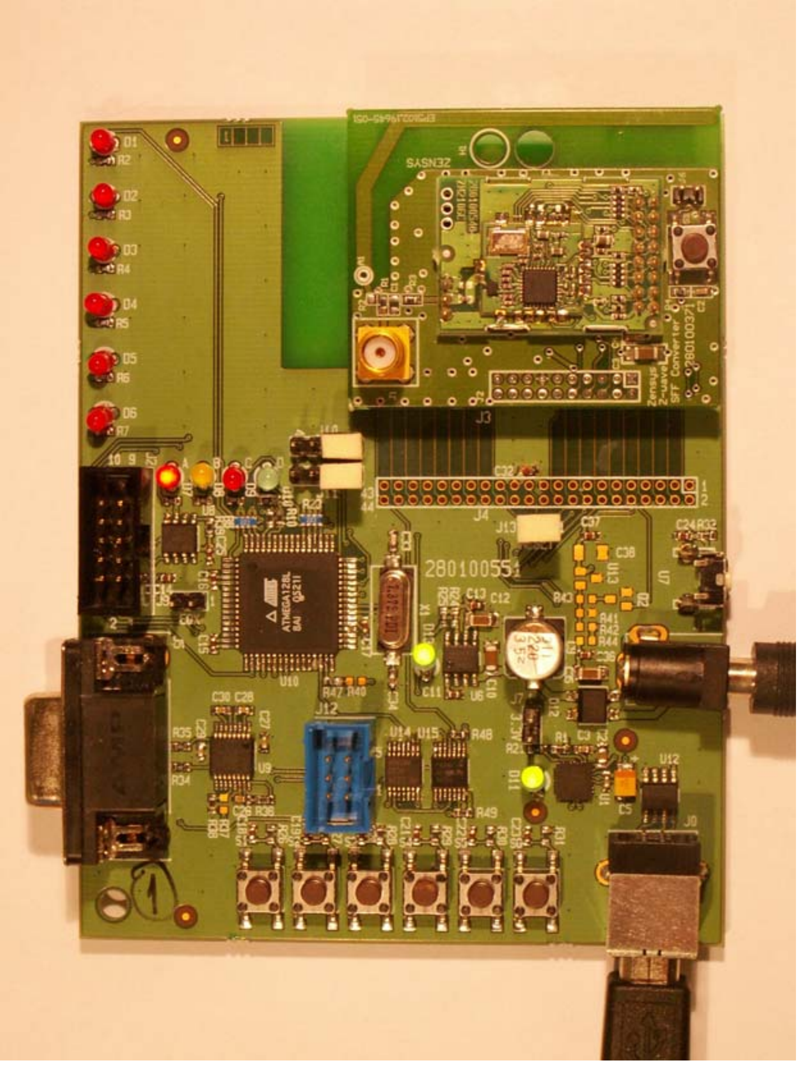


Figure 2. The ZDP02A Z-Wave Programmer with a Z-Wave module mounted.

The figure below show another configuration of the Z-Wave Programmer HW with a Z-Wave module connected via the ISP cable. The Z-Wave Module can be powered via the Z-Wave Programmer HW. It is not possible to write data to the external EEPROM on the Z-Wave module in this configuration.

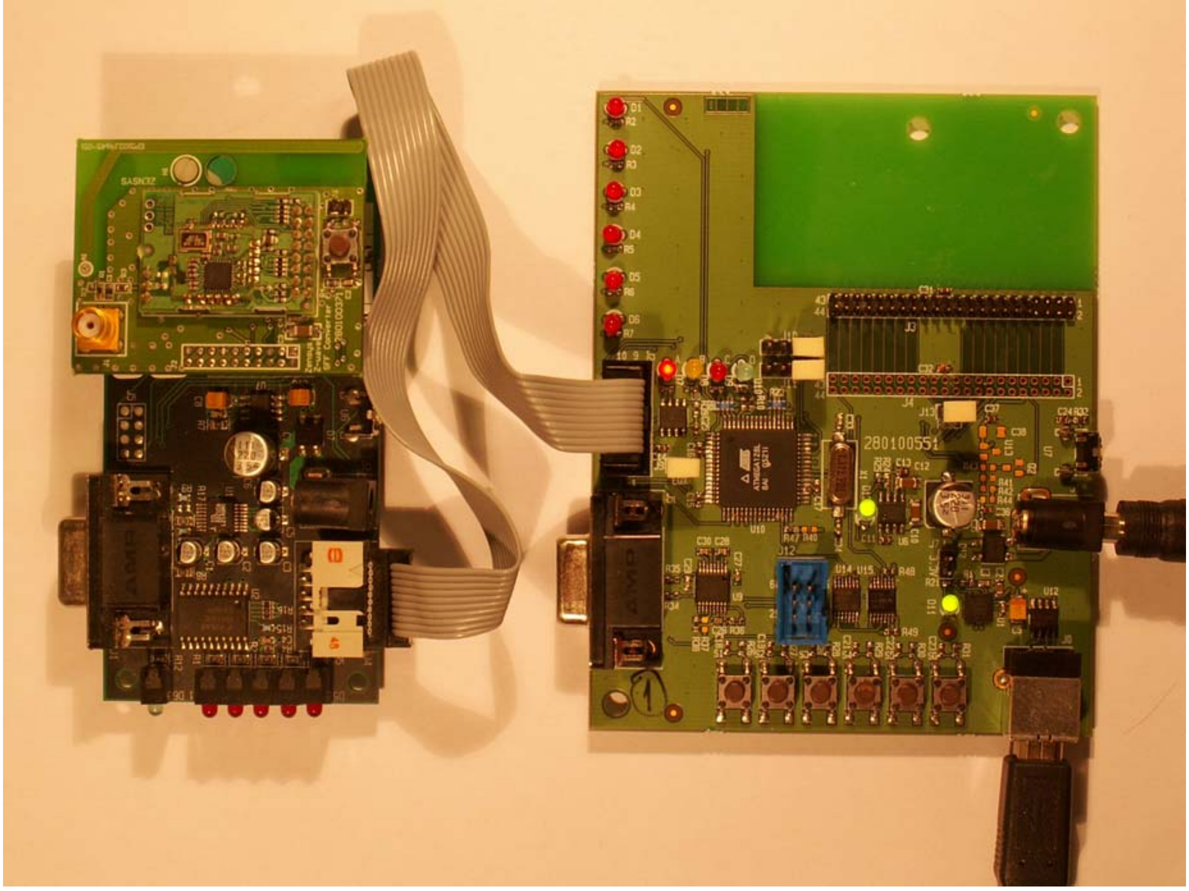


Figure 3. The ZDP02A Z-Wave Programmer with a Z-Wave module connected via the ISP cable.

3.1.3 Hardware installation using ZDP03A Z-Wave Programmer

The ZDP02A Z-Wave Programmer can be used for programming the flash in the Z-Wave 100/200/300 Series Single Chips and the OTP in the Z-Wave 400 Series Single Chips. It is also possible to program the external EEPROM on the Z-Wave module except 100 Series based modules when using ZDP03A. The ZDP03A Z-Wave Programmer can be powered via USB interface as well as using external power supply. The ZDP03A Z-Wave Programmer is shown on the figure below:

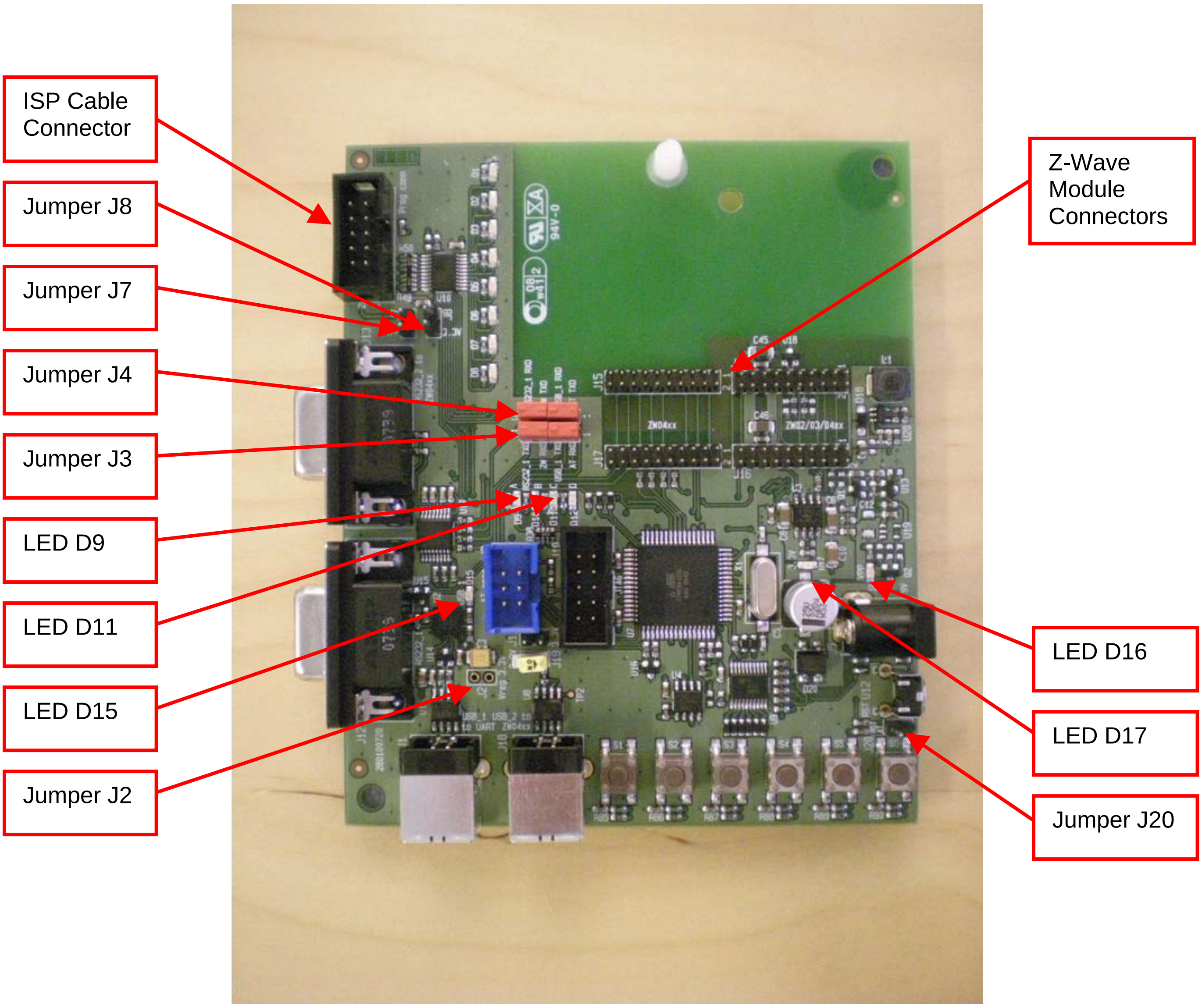


Figure 4. The ZDP03A Z-Wave Programmer connectors, LEDs and jumpers layout.

Jumpers J3 and J4 must be shorted as shown on Figure 4.

Jumper J20 must be off. When ON it reset the ATmega128 and LED D9 turns off.

Jumper J8 must be shorted in case the Z-Wave module connected via the ISP cable is powered (3.3V) from the Z-Wave programmer. Jumper J8 must be open in case the Z-Wave module connected via the ISP cable uses an independent power-supply. Jumper J7 must be shorted to enable OTP programming on the Z-Wave module when connected via the ISP cable.

Jumper J2 must be shorted in case the Z-Wave programmer/module is powered via the USB interface.

IMPORTANT: Check carefully that the jumpers are mounted correctly on the Z-Wave Programmer HW.

Several LED’s are used to indicate the state of the programmer. LED15 is on if USB controller power is on and there is a connection to the host PC. LED17 is on if the board is powered on. LED16 is on if the board is powered with an external power supply. LED9 is on when the Z-Wave Programmer firmware is running. LED D11 is on when the Z-Wave Programmer is busy and it is off when the programmer is idle.

The figure below shows the ZDP03A Z-Wave Programmer HW with a Z-Wave module mounted in the Z-Wave Module connector. The Z-Wave Programmer is connected to the PC using the USB interface and powered by an external power supply. The module can be mounted in the Z-Wave Module connector while debugging the embedded application.

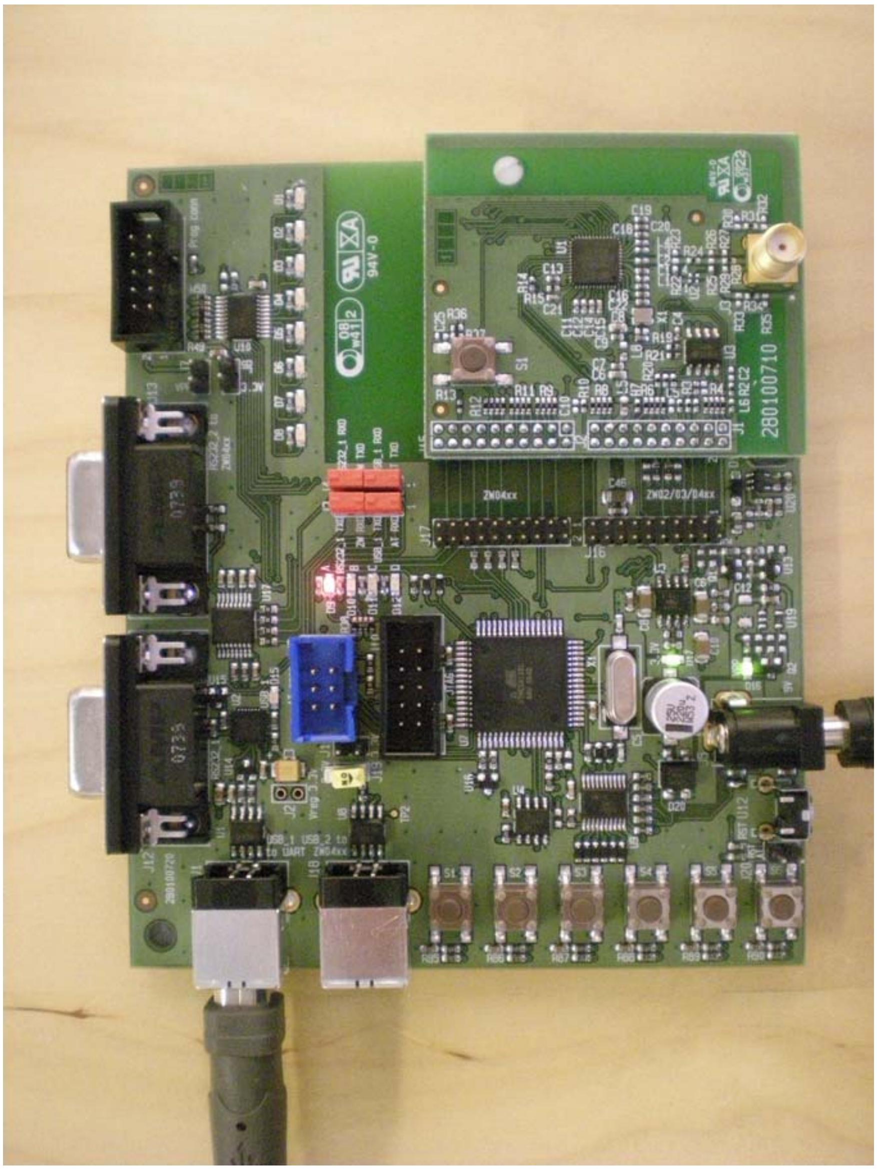


Figure 5. The ZDP03A Z-Wave Programmer with a Z-Wave module mounted.

The figure below show another configuration of the Z-Wave Programmer HW with a Z-Wave module connected via the ISP cable. The Z-Wave Module can be powered via the Z-Wave Programmer HW. It is not possible to write data to the external EEPROM on the Z-Wave module in this configuration.

Notice: It is not possible to program OTP memory on the ZM4225 when mounted on the ZW0x0x Interface Module.

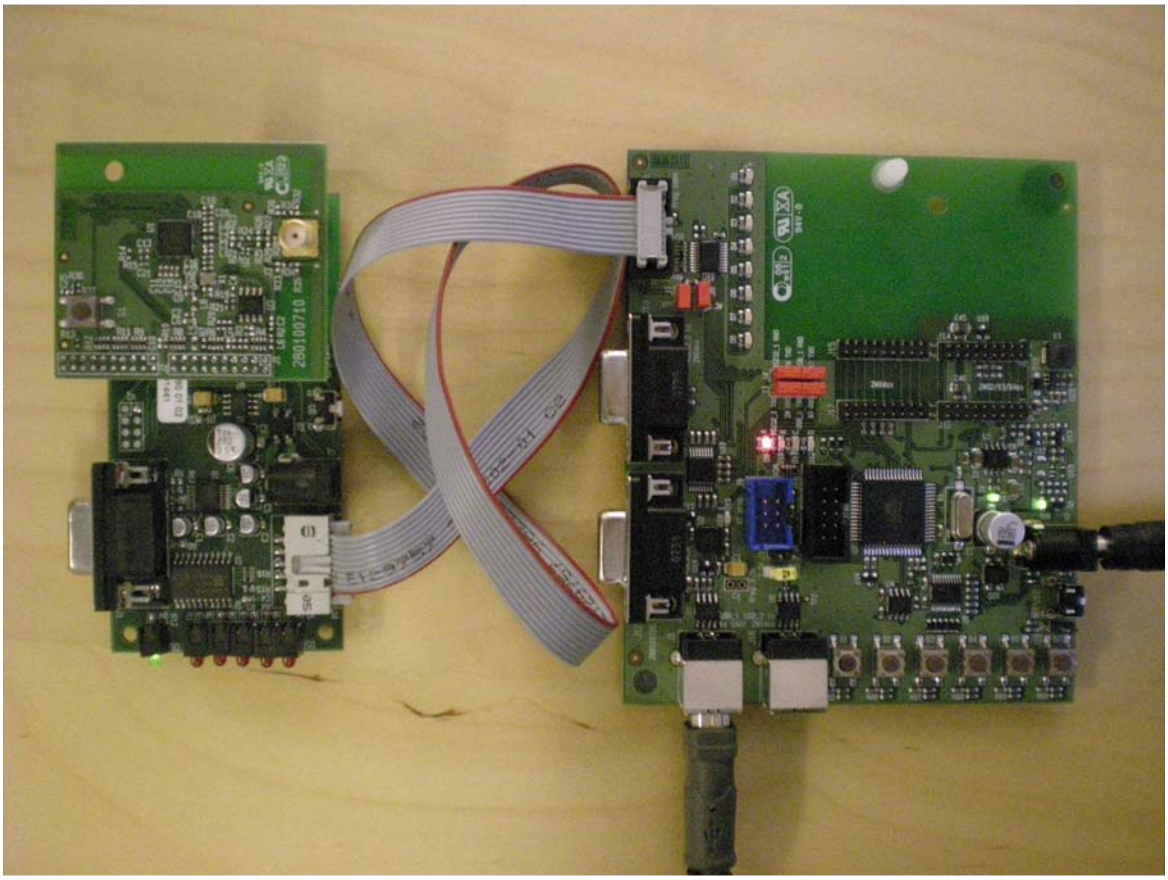
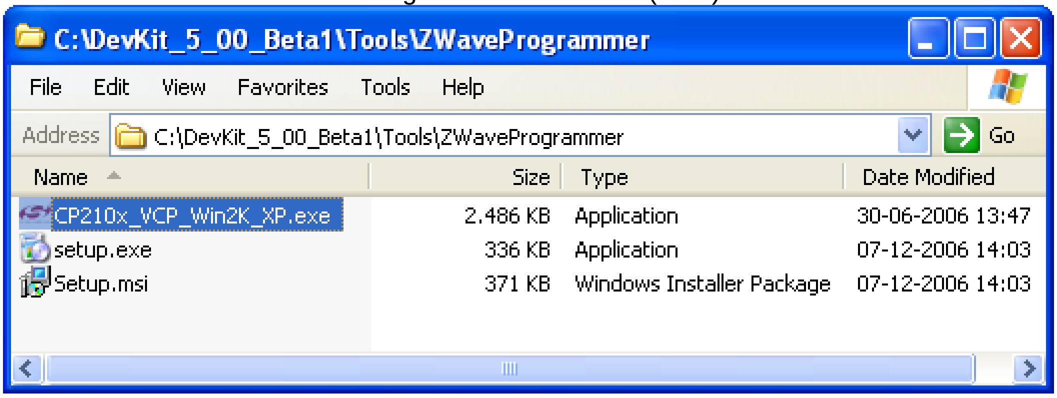


Figure 6. The ZDP03A Z-Wave Programmer with a Z-Wave module connected via the ISP cable.

3.1.4 USB Driver installation

It is essential to follow the proper hardware installation procedure to ensure that your Z-Wave Programmer software works correct.

1. Install CP210x USB to UART Bridge Virtual COM Port (VCP) driver.



C:\DevKit_5_00_Beta1\Tools\ZWaveProgrammer

Address C:\DevKit_5_00_Beta1\Tools\ZWaveProgrammer

Name	Size	Type	Date Modified
CP210x_VCP_Win2K_XP.exe	2.486 KB	Application	30-06-2006 13:47
setup.exe	336 KB	Application	07-12-2006 14:03
Setup.msi	371 KB	Windows Installer Package	07-12-2006 14:03

Figure 7. USB driver location

Note! This should be done ***before*** the Z-Wave Programmer hardware (ZDP02A Z-Wave Development Platform) is connected to the USB interface.
To run the CP210x VCP driver installation wizard, open C:\DevKit_X_YY\Tools\Programmer\CP210x_VCP_Win2K_XP.exe executable file and follow the installation wizard.

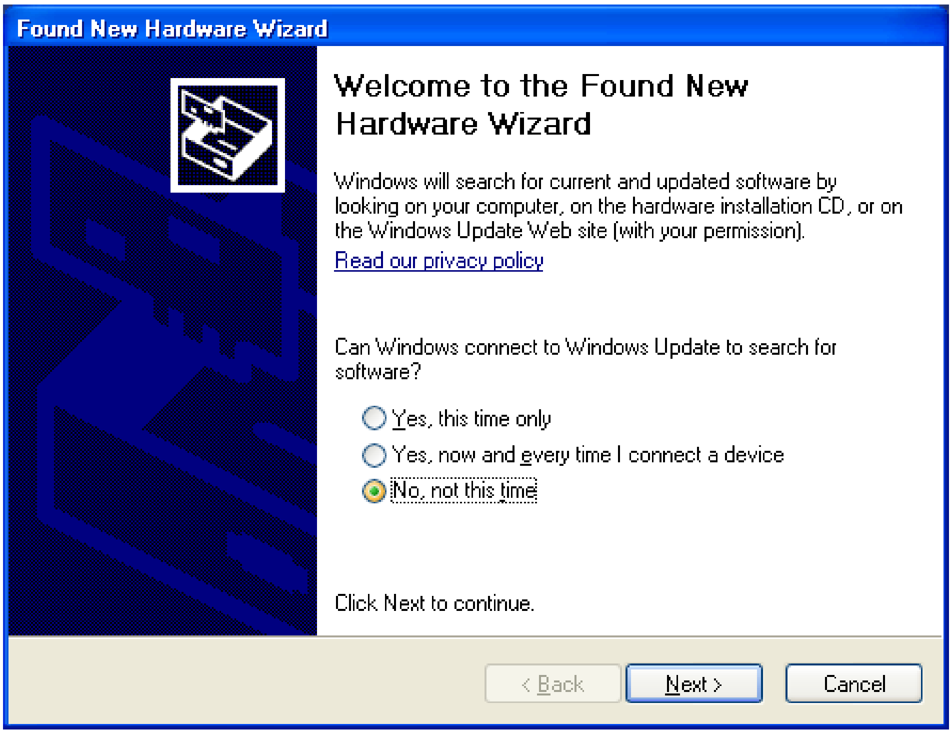
2. Check the jumper settings on Z-Wave Programmer hardware.

3. Mount Z-Wave module either directly in Z-Wave module connector or connect units using the ISP cable.

   Z-Wave module can be mounted directly on Z-Wave Programmer hardware (ZDP02A Z-Wave Development Platform) using the Z-Wave module connector. This configuration allows writing data to the external EEPROM on the Z-Wave module and debugging the embedded application. Alternatively Z-Wave module can be mounted on Z-Wave interface unit and then connected to the Z-Wave Programmer hardware via the ISP cable. The Z-Wave module can be powered via the Z-Wave Programmer hardware. In this configuration it is impossible to write data to the external EEPROM on the Z-Wave module.

4. Plug the power supply.

5. Connect Z-Wave Programmer hardware (ZDP02A Z-Wave Development Platform) to the PC using the USB cable.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only

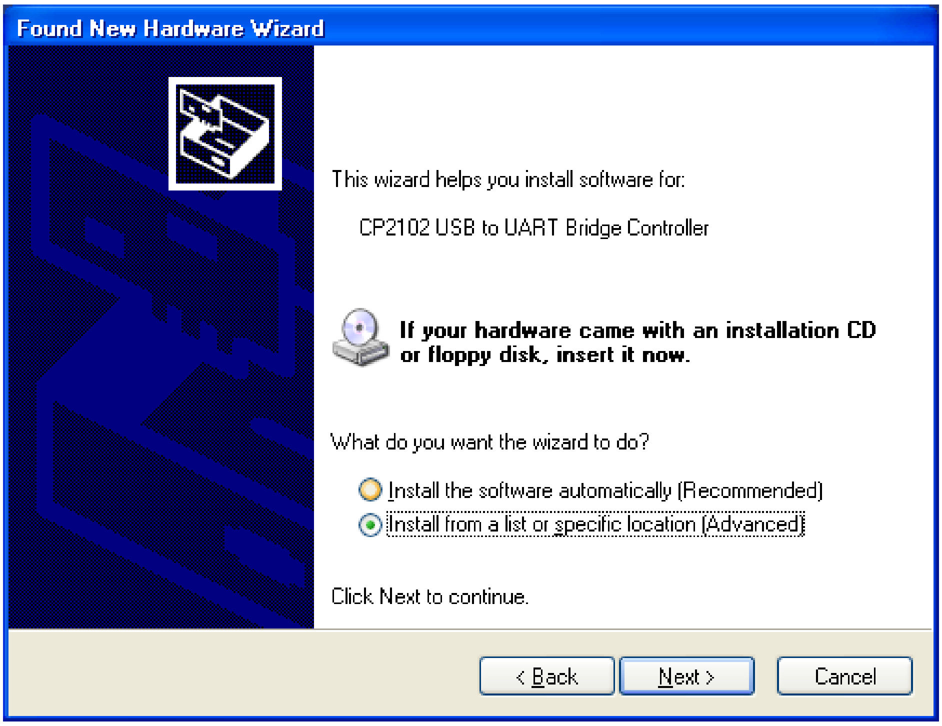
○ Yes, now and every time I connect a device

● No, not this time

Click Next to continue.

Figure 8. Z-Wave Programmer HW installation step 1

Windows will detect the new hardware and the new hardware wizard will start.
Select **No, not this time** on the wizard then click **Next**.



Found New Hardware Wizard

This wizard helps you install software for:

CP2102 USB to UART Bridge Controller

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)

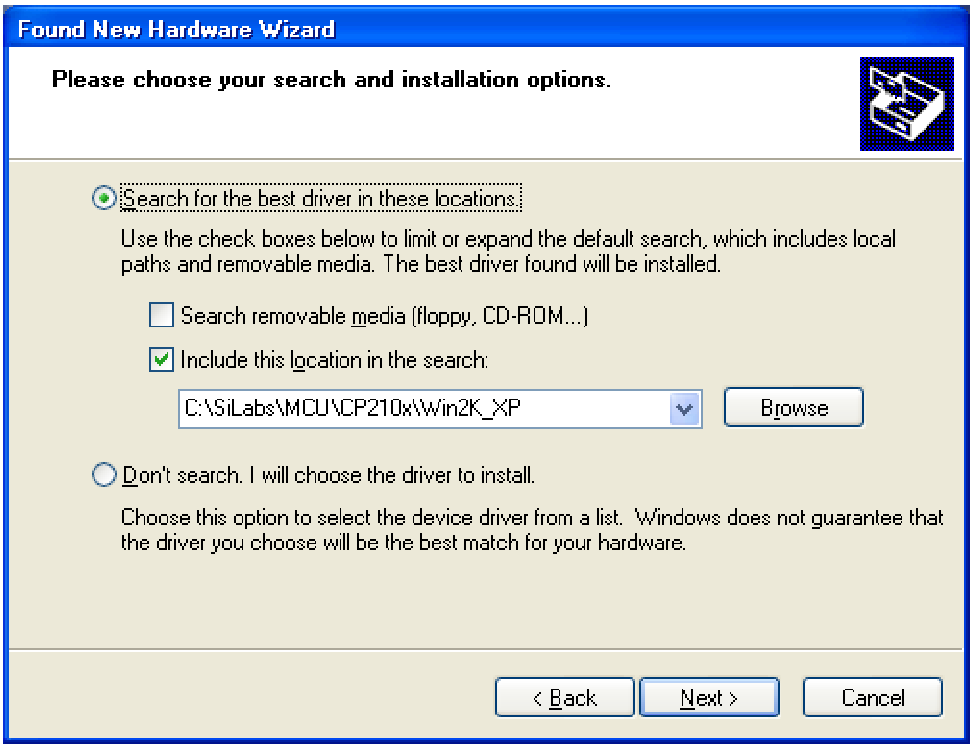
● Install from a list or specific location (Advanced)

Click Next to continue.

Figure 9. Z-Wave Programmer HW installation step 2

On the next page select **Install from a list or specific location** option.
On the next Wizard page point to the location of the USB driver you installed previously at step 1,

it is "C:\SiLabs\MCU\CP210x\Windows_2K_XP_S2K3_Vista" folder by default.



Found New Hardware Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

□ Search removable media (floppy, CD-ROM...)

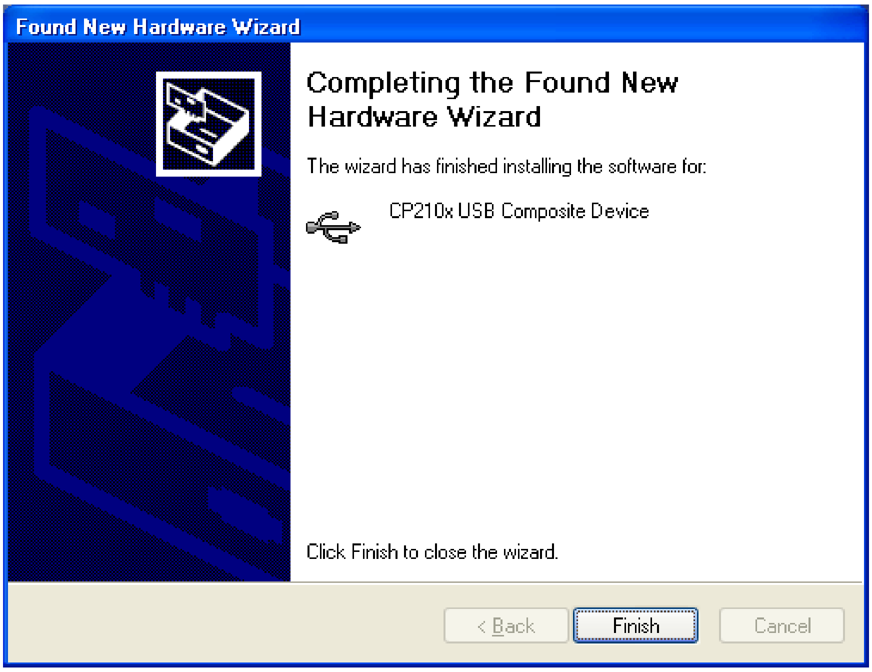
☑ Include this location in the search:

C:\SiLabs\MCU\CP210x\Win2K_XP

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Figure 10. Z-Wave Programmer installation



Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

CP210x USB Composite Device

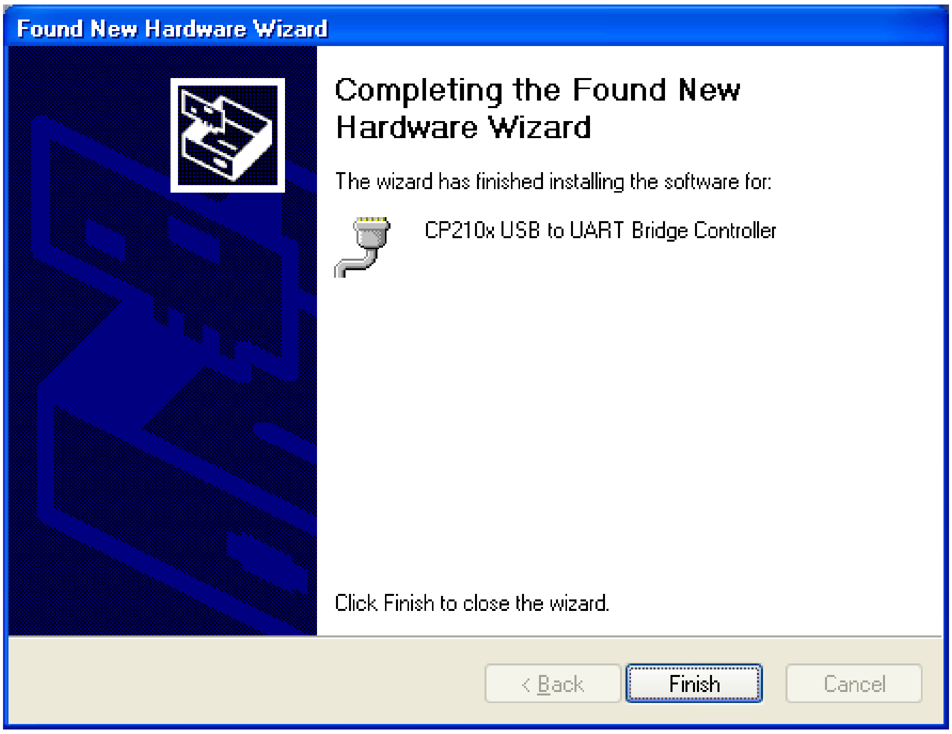
Click Finish to close the wizard.

Figure 11. End of the USB driver installation part 1

6. Install driver for the new hardware.

   Follow the wizard to install CP210x USB to UART Bridge Virtual COM Port (VCP) driver as

described above.
Note! This step is required to complete ***twice*** to avoid being prompted ***every time*** you connect the Z-Wave Programmer hardware.



Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

CP210x USB to UART Bridge Controller

Click Finish to close the wizard.

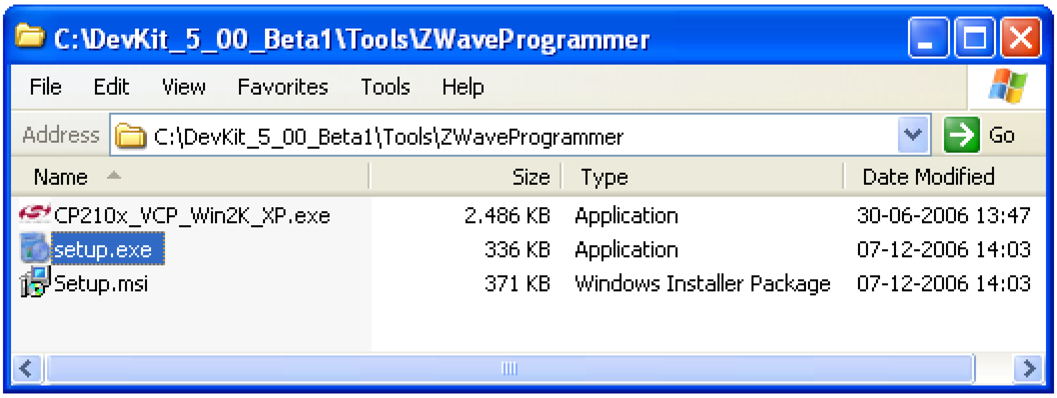
Figure 12. End of the USB driver installation part 2

7. Restart the PC when the USB driver installation is finished.

Attention! Make sure you completed this procedure to install the Z-Wave Programmer hardware.

3.1.5 Z-Wave Programmer software installation

The Z-Wave Programmer software is installed by browsing to the C:\DevKit_X_YY\Tools\ZWaveProgrammer\ directory and double click on the **setup.exe** file.

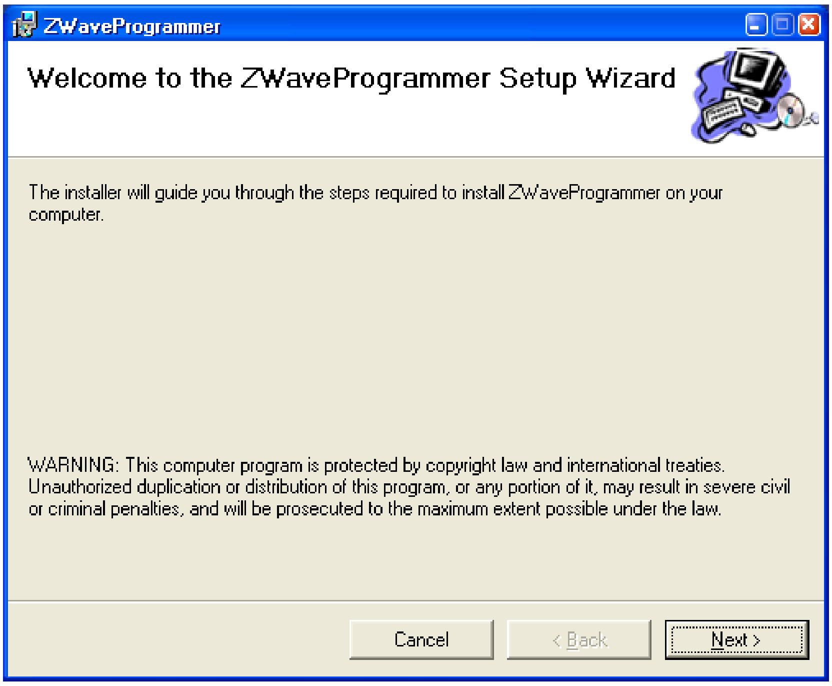

C:\DevKit_5_00_Beta1\Tools\ZWaveProgrammer

Address: C:\DevKit_5_00_Beta1\Tools\ZWaveProgrammer

Name	Size	Type	Date Modified
CP210x_VCP_Win2K_XP.exe	2.486 KB	Application	30-06-2006 13:47
setup.exe	336 KB	Application	07-12-2006 14:03
Setup.msi	371 KB	Windows Installer Package	07-12-2006 14:03

Figure 13. Execution of setup.exe

Click on the '**Next>**' button in the setup wizard


ZWaveProgrammer

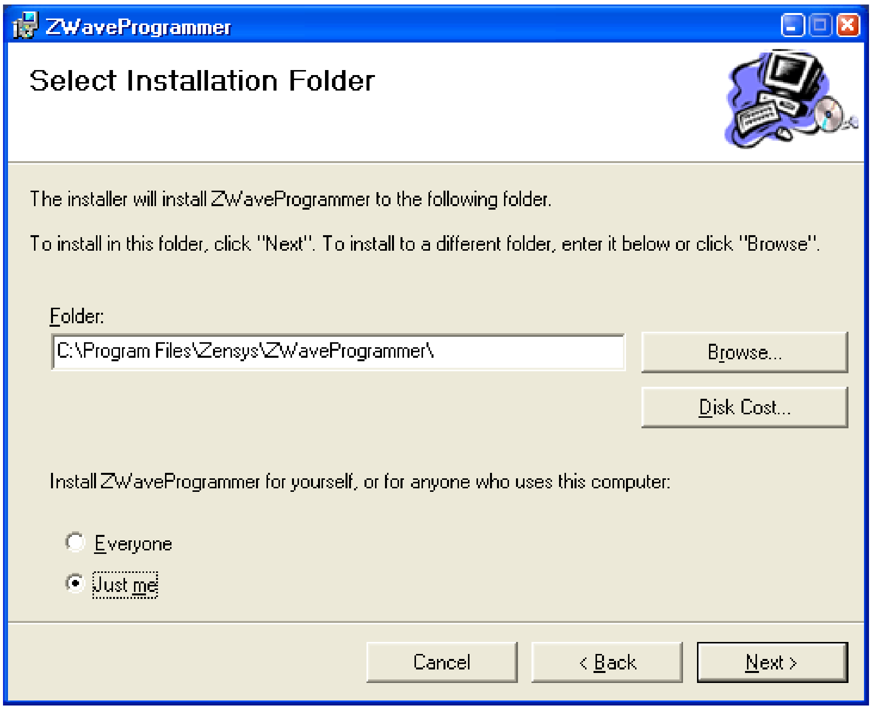
Welcome to the ZWaveProgrammer Setup Wizard

The installer will guide you through the steps required to install ZWaveProgrammer on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure 14. Start setup wizard

Select new location to install the program on it or leave the default settings then click on the ‘**Next>**’ button.



ZWaveProgrammer

Select Installation Folder

The installer will install ZWaveProgrammer to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder:
C:\Program Files\Zensys\ZWaveProgrammer\

Install ZWaveProgrammer for yourself, or for anyone who uses this computer:

○ Everyone
● Just me

Figure 15. Select installation folder

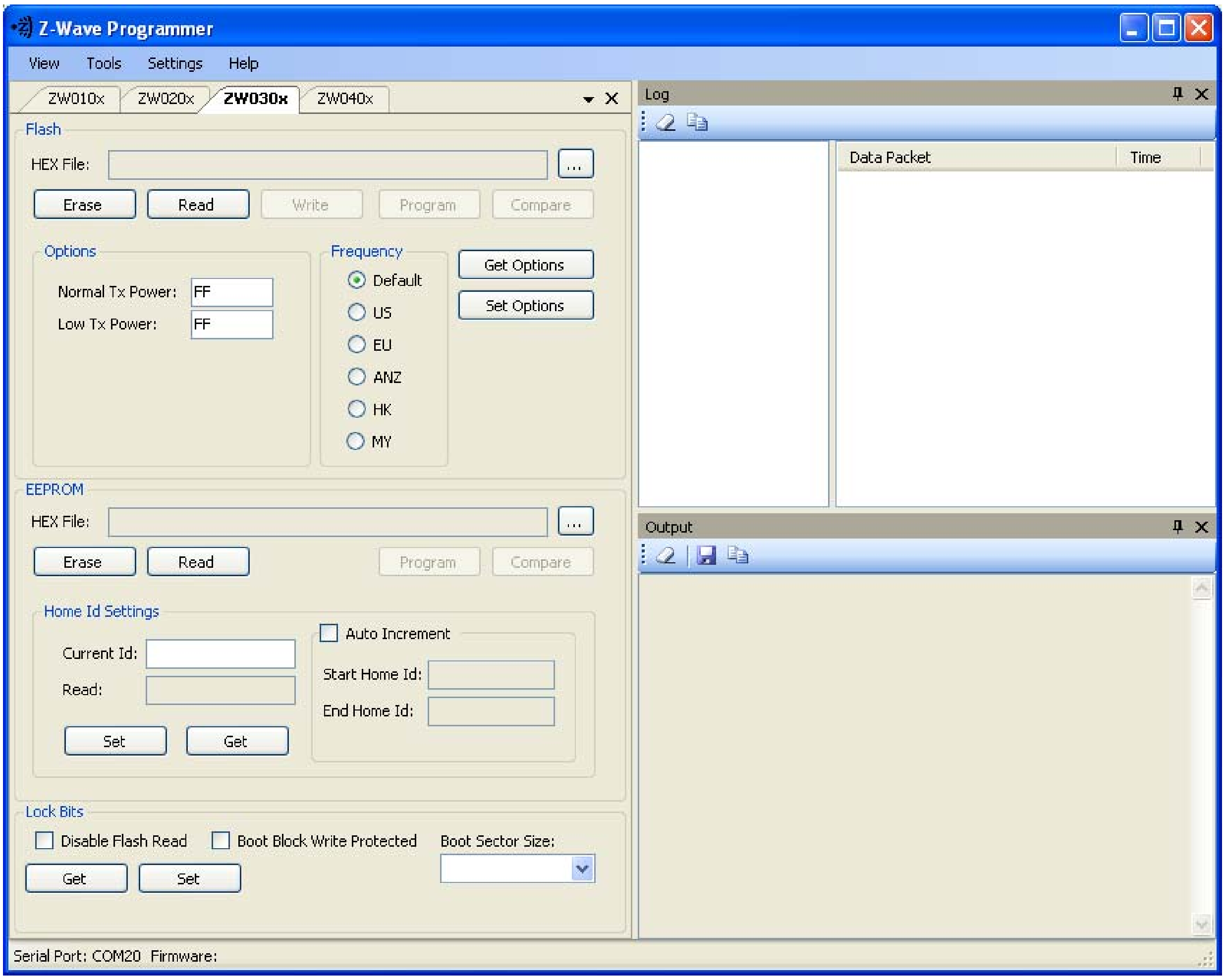
Click again on the ‘**Next>**’ button and the wizard will start installing the Z-Wave Programmer software. Finally click on the ‘**Close**’ button to exit the installation wizard.

Run the Z-Wave Programmer from the start menu under:

Start->All Programs->Zensys->ZWaveProgrammer.

4 OVERVIEW

A Windows GUI and console interfaces are provided to manage the programming of FLASH and EEPROM. Figure 16 below shows a screen dump of the Z-Wave Programmer GUI.



Normal Tx Power: FF

Low Tx Power: FF

● Default

Serial Port: COM20 Firmware:

Figure 16. Z-Wave Programmer main window

The Windows GUI has the following views:

4.1 Main Menu

The Main Menu consists of the following items:

Menu item	Description
View	Offers selection of sections to be displayed, and also comprises the Exit function.
Tools	Comprises **Detect Target, Upgrade Firmware** and **Reset Z-Wave Module** options.
Settings	Comprises options for **COM Port selection**, **Detect Target on Startup**, **Interface Filter**, and **Show HEX files path.**
Help	Comprises links to the integrated Help Pane, **API References**, and the **About** link.

4.2 Log

The **Log** view keeps the log of application's actions.

4.3 Output

The **Output** view shows the content of the HEX file after the **Read** functionality has been activated. From here, it is possible to save the obtained HEX file.

4.4 The ZW010x, ZW020x and ZW030x tabs

These tabs have a unified outlook as on Figure 16.

4.4.1 Flash

The **Flash** section comprises the following controls:

- **HEX File** location bar;
- **Erase** button (to erase the flash);
- **Read** button (to read the flash contents);
- **Write** button (to write HEX files contents to the flash);
- **Program** button (to write and verify HEX files contents to the flash);
- **Compare** button (to compare the flash contents with a hex file);
- **Normal TX Power** and **Low TX Power** textbox controls;
- **Frequency** selection radiobutton control;
- **Get Options** button (to obtain current TX power and RF settings);
- **Set Options** button (to write the TX power and RF settings to the flash)

4.4.2 EEPROM

The **EEPROM** section comprises the following controls:

- **HEX File** location bar;
- **Erase** button (to clear the EEPROM);
- **Read** button (to read the EEPROM contents);
- **Program** button (to write HEX files contents to the EEPROM);
- **Compare** button (to compare the EEPROM contents with a file);
- **Home ID settings** (to configure Home ID of the module)

NOTICE: Not possible to program external EEPROM on 100 Series based modules when using ZDP03A.

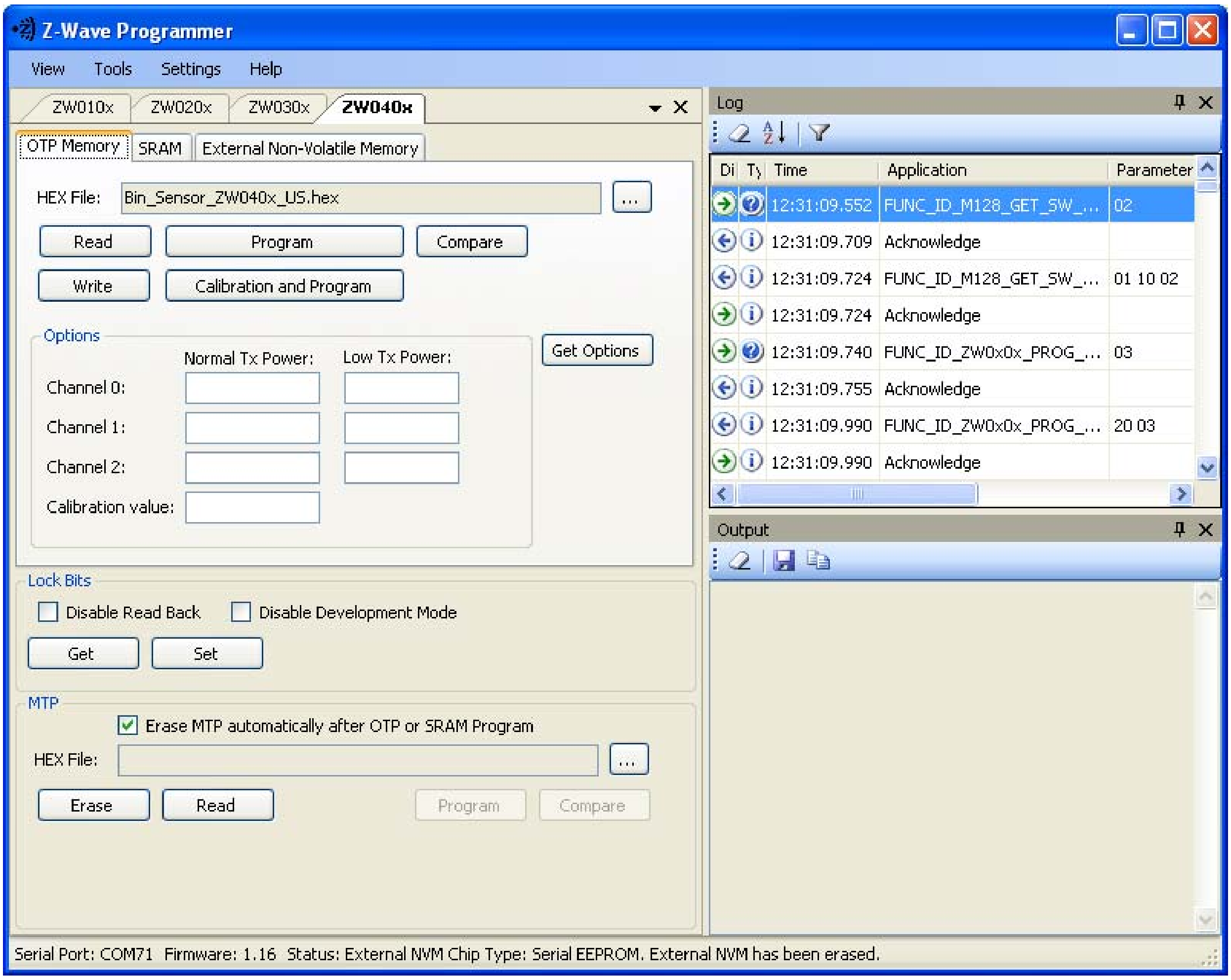
4.4.3 Lock Bits

The Lock Bits section comprises the following controls:

- **Disable Flash Read** checkbox;
- **Boot Block Write Protected** checkbox;
- **Boot Sector Size** drop list;
- **Get** button;
- **Set** button

4.5 The ZW040x tab

4.5.1 OTP (One-Time-Programmable) Memory



HEX File: Bin_Sensor_ZW040x_US.hex

□ Disable Read Back □ Disable Development Mode

☑ Erase MTP automatically after OTP or SRAM Program

HEX File:

Di	Ty	Time	Application	Parameter
→	?	12:31:09.552	FUNC_ID_M128_GET_SW_...	02
←	i	12:31:09.709	Acknowledge	
←	i	12:31:09.724	FUNC_ID_M128_GET_SW_...	01 10 02
→	i	12:31:09.724	Acknowledge	
→	?	12:31:09.740	FUNC_ID_ZW0x0x_PROG_...	03
←	i	12:31:09.755	Acknowledge	
←	i	12:31:09.990	FUNC_ID_ZW0x0x_PROG_...	20 03
→	i	12:31:09.990	Acknowledge	

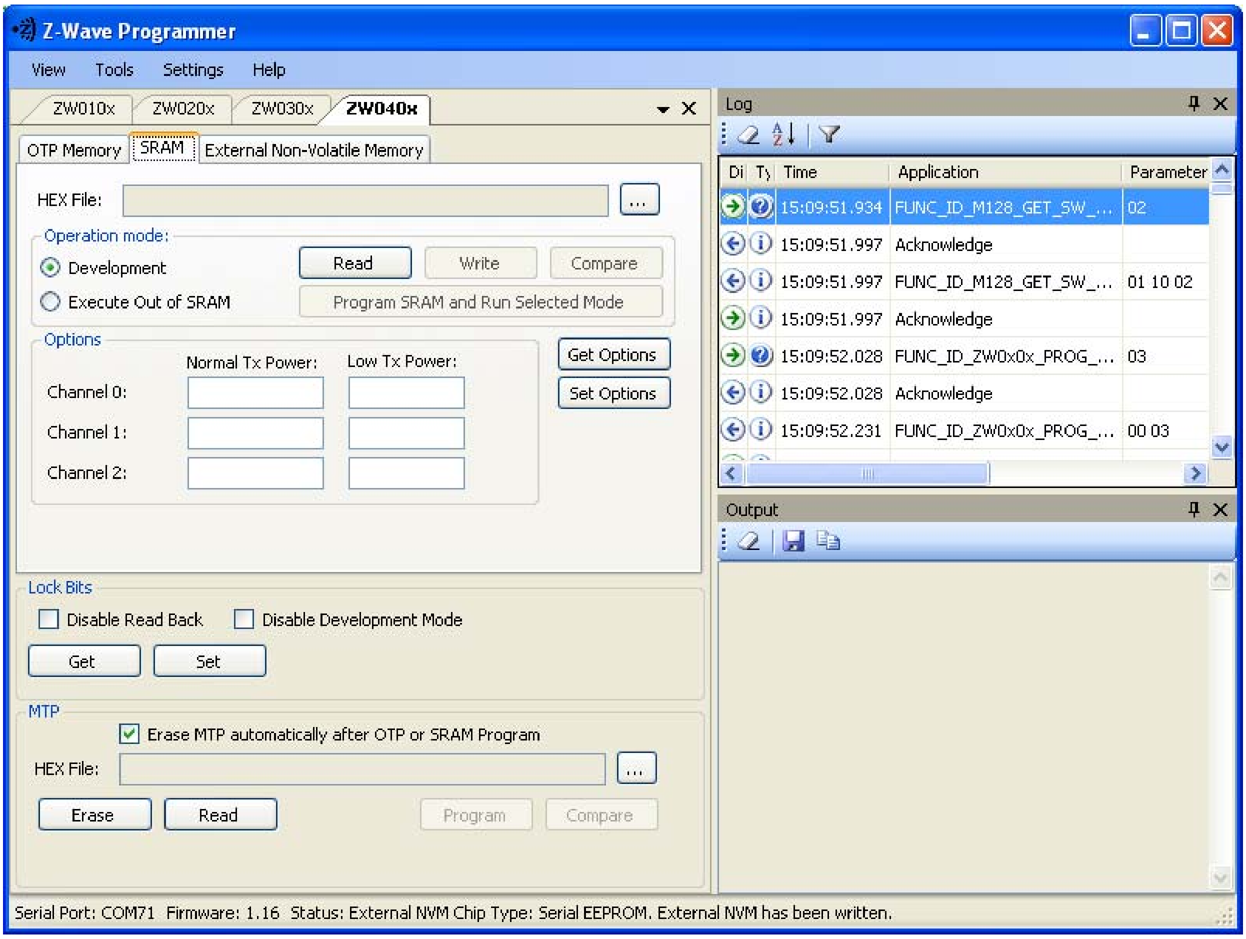
Serial Port: COM71 Firmware: 1.16 Status: External NVM Chip Type: Serial EEPROM. External NVM has been erased.

Figure 17. The ZW040x tab

The **OTP Memory** section includes:

- **HEX File** location bar;
- **Read** button (to read the OTP memory contents);
- **Write** button (to write HEX files contents to the OTP memory);
- **Program** button (to write and verify HEX files contents to the OTP memory);
- **Compare** button (to compare the flash contents with a hex file);
- **Calibration and Program** button (to perform calibration and then program the chip)
- **Normal TX Power** and **Low TX Power** textbox controls;
- **Get Options** button (to obtain current TX power and RF settings);
- **Lock Bits,** the same as in ZW010x-ZW030x tabs.

4.5.2 SRAM



● Development
○ Execute Out of SRAM

☑ Erase MTP automatically after OTP or SRAM Program

Di	Ty	Time	Application	Parameter
		15:09:51.934	FUNC_ID_M128_GET_SW_...	02
		15:09:51.997	Acknowledge	
		15:09:51.997	FUNC_ID_M128_GET_SW_...	01 10 02
		15:09:51.997	Acknowledge	
		15:09:52.028	FUNC_ID_ZW0x0x_PROG_...	03
		15:09:52.028	Acknowledge	
		15:09:52.231	FUNC_ID_ZW0x0x_PROG_...	00 03

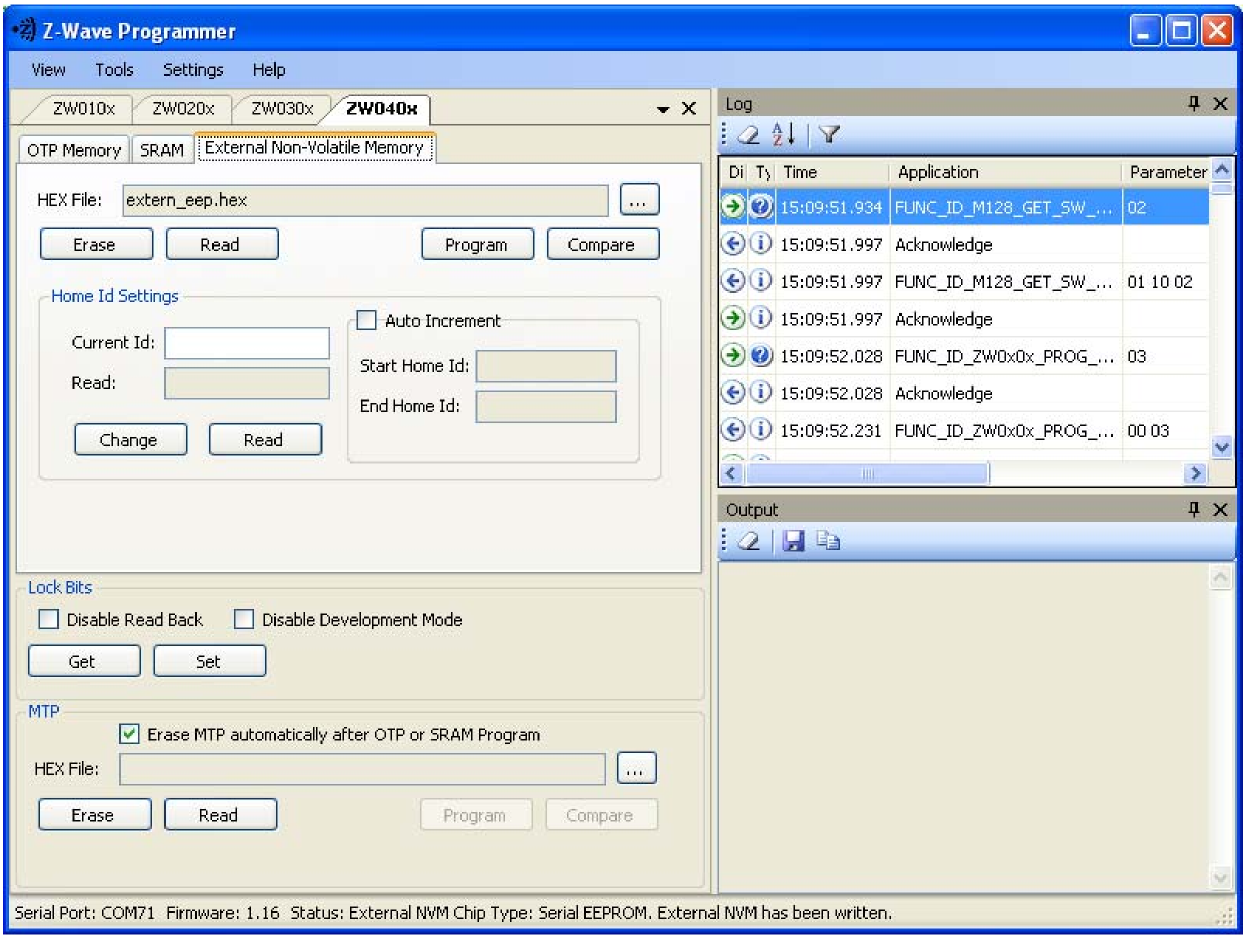
Serial Port: COM71 Firmware: 1.16 Status: External NVM Chip Type: Serial EEPROM. External NVM has been written.

Figure 18. SRAM

The **SRAM** section includes:

- **HEX File** location bar;
- **Operation Mode** radiobutton control: **Development**, **Execute out of SRAM**, with following subcontrols:
    - **Read** button (to read the SRAM contents);
    - **Write** button (to write HEX files contents to the SRAM);
    - **Compare** button (to compare the SRAM contents with a hex file);
    - **Program SRAM and Run Selected Mode** button;
    - **Options** section.

4.5.3 External Non-Volatile Memory



HEX File: extern_eep.hex

□ Auto Increment

□ Disable Read Back □ Disable Development Mode

☑ Erase MTP automatically after OTP or SRAM Program

Time	Application	Parameter
15:09:51.934	FUNC_ID_M128_GET_SW_...	02
15:09:51.997	Acknowledge	
15:09:51.997	FUNC_ID_M128_GET_SW_...	01 10 02
15:09:51.997	Acknowledge	
15:09:52.028	FUNC_ID_ZW0x0x_PROG_...	03
15:09:52.028	Acknowledge	
15:09:52.231	FUNC_ID_ZW0x0x_PROG_...	00 03

Serial Port: COM71 Firmware: 1.16 Status: External NVM Chip Type: Serial EEPROM. External NVM has been written.

Figure 19. External NVM

The **External Non-Volatile Memory** section includes:

- **HEX File** location bar;
- **Erase, Read, Program, Compare** buttons;
- **Home ID Settings** control.
-

4.5.4 MTP

The **MTP** section includes standard controls for HEX file managing: the HEX file location bar, and the buttons to **Erase**, **Read**, **Program** and **Compare** the MTP content with the HEX file.

5 PROGRAMMING

5.1 Programming setup

The programming setup which is depicted in Figure 20 consists of

- The Z-Wave Programmer HW.
- A Single Chip module mounted in the Z-Wave module socket (alternatively connected via the ISP cable).
- A PC running Z-Wave Programmer GUI on windows XP.
- USB cable.
- One power supply connected to the programmer.

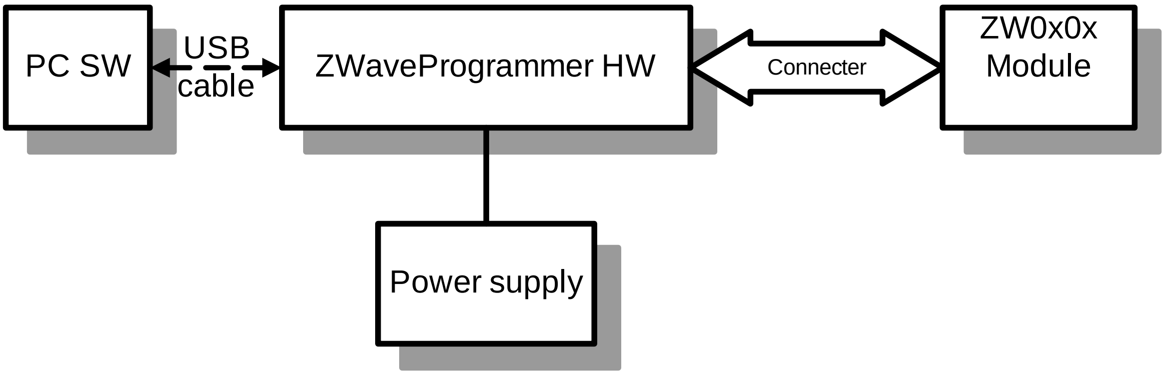


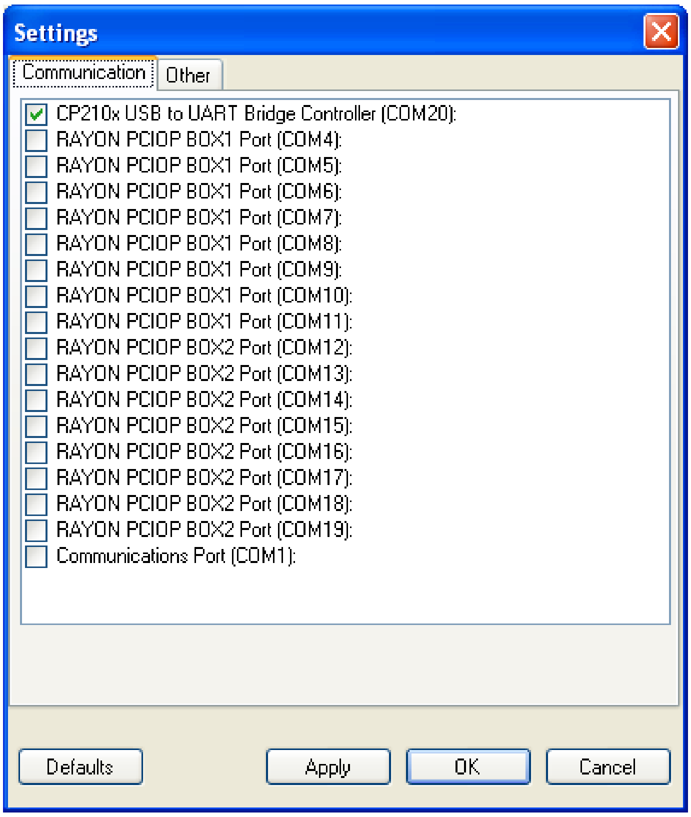
Figure 20. Programming Setup

5.2 Configuration

5.2.1 Selecting a COM port

First select the virtual COM port assigned to the Z-Wave Programmer. In the Settings menu item:

Open Settings->COM Port and select the COM port (COM20 in this example)



☑ CP210x USB to UART Bridge Controller (COM20):
□ RAYON PCIOP BOX1 Port (COM4):
□ RAYON PCIOP BOX1 Port (COM5):
□ RAYON PCIOP BOX1 Port (COM6):
□ RAYON PCIOP BOX1 Port (COM7):
□ RAYON PCIOP BOX1 Port (COM8):
□ RAYON PCIOP BOX1 Port (COM9):
□ RAYON PCIOP BOX1 Port (COM10):
□ RAYON PCIOP BOX1 Port (COM11):
□ RAYON PCIOP BOX2 Port (COM12):
□ RAYON PCIOP BOX2 Port (COM13):
□ RAYON PCIOP BOX2 Port (COM14):
□ RAYON PCIOP BOX2 Port (COM15):
□ RAYON PCIOP BOX2 Port (COM16):
□ RAYON PCIOP BOX2 Port (COM17):
□ RAYON PCIOP BOX2 Port (COM18):
□ RAYON PCIOP BOX2 Port (COM19):
□ Communications Port (COM1):

Figure 21. Selecting COM port for the Programmer

The selected COM port is used as the default port every time the Z-Wave Programmer is started until it is changed again.

5.2.2 Detecting target

Set the Z-Wave Programmer to detect the target type during Z-Wave Programmer startup. To do this, from the main menu, select **Tools -> Detect Target** as shown in Figure 22.

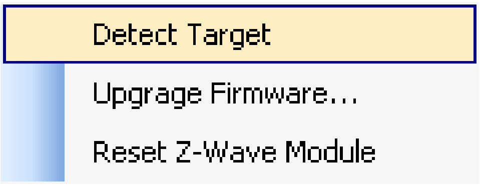
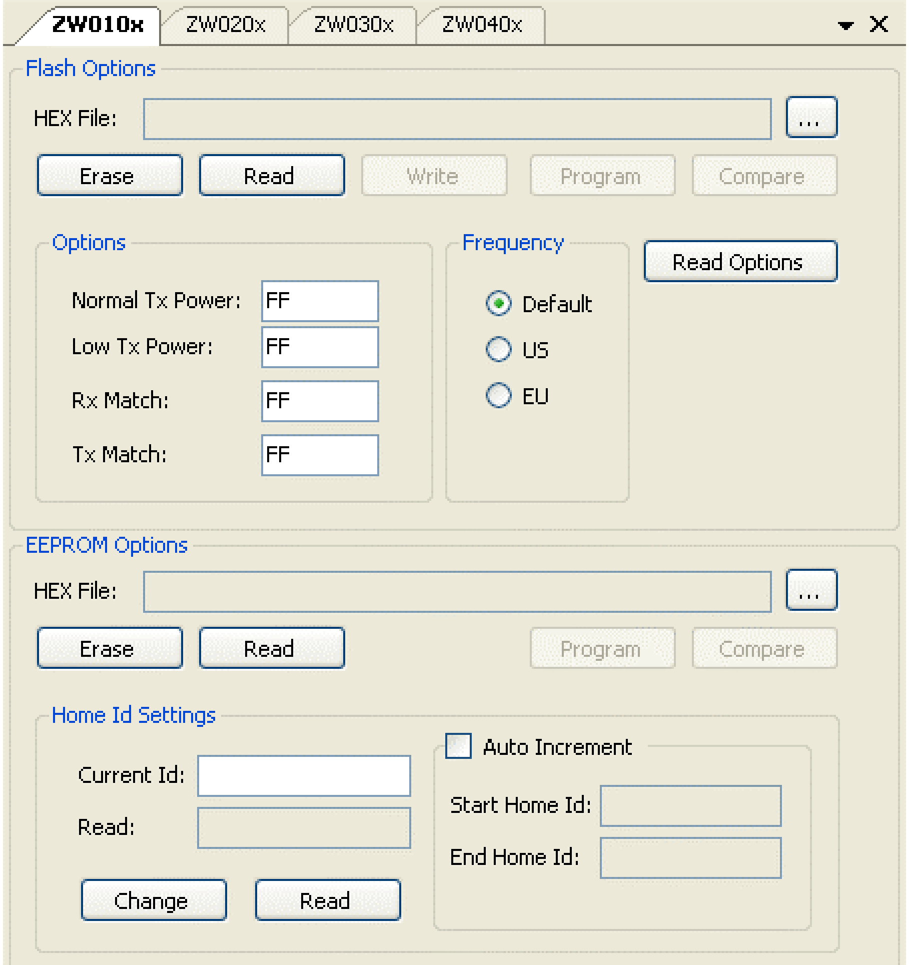


Figure 22. Detect target

5.3 Flash / OTP Memory handling

5.3.1 Writing a HEX file to Flash/OTP Memory

Click on the open file icon next to the hex file edit box field as shown in Figure 23. An open file dialog box will appear, browse to the desired HEX file and select it. Next select the Z-Wave Single Chip tab that matches the chip to program, in our example ZW0201 Single Chip. Refer to [1] regarding the available sample applications and the associated HEX files. Remember that the external EEPROM on the Z-Wave module must be initialized properly; otherwise the protocol or application can get corrupted data as input and possibly fail. The enhanced slave and all the controller libraries use the external EEPROM.



Normal Tx Power: FF

Low Tx Power: FF

Rx Match: FF

Tx Match: FF

● Default

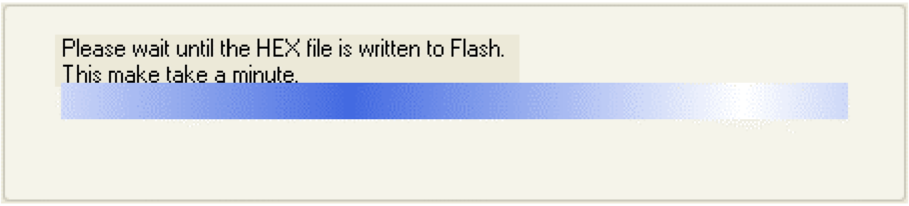
○ US

○ EU

□ Auto Increment

Figure 23. FLASH and EEPROM files open icons

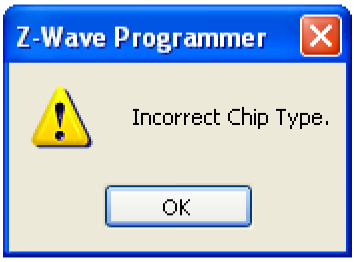
To Write and verify the content of the HEX file to the Flash/OTP Memory, click on the '**Program**' button in the **Flash Options / OTP Memory** group box. Click on the '**Write**' button to write the HEX file content to the flash. After one click on the button the programming process starts. In case of Flash, the process starts by erasing the flash. After the flash is erased, the Z-Wave Programmer will start to write data to flash as shown in Figure 24.



Please wait until the HEX file is written to Flash.
This make take a minute.

Figure 24. Flash page writing

When the writing process is completed, a “**Program Done**” is shown in the status. If any error occurs during programming, a failure message will appear at the bottom of the dialog box as shown in Figure 25.



Z-Wave Programmer

Incorrect Chip Type.

Figure 25. Write failure

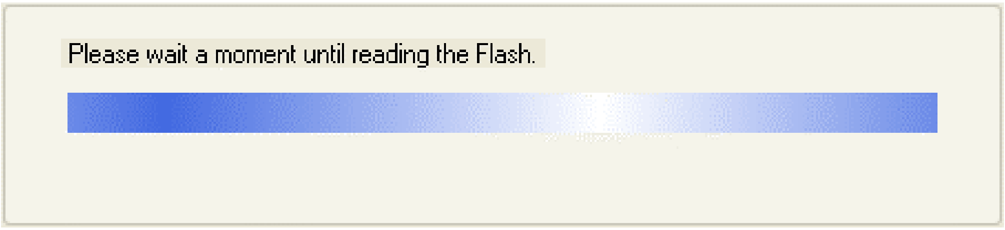
5.3.2 Erasing the flash

To erase the entire flash, simply click on the ‘**Erase’** button in the **Flash Options /OTP Memory** group box.

5.3.3 Reading the Flash/OTP Memory content

The Z-Wave Single Chip flash/OTP content can be saved to a file. To generate a file with the content of the Z-Wave Single Chip do the following:

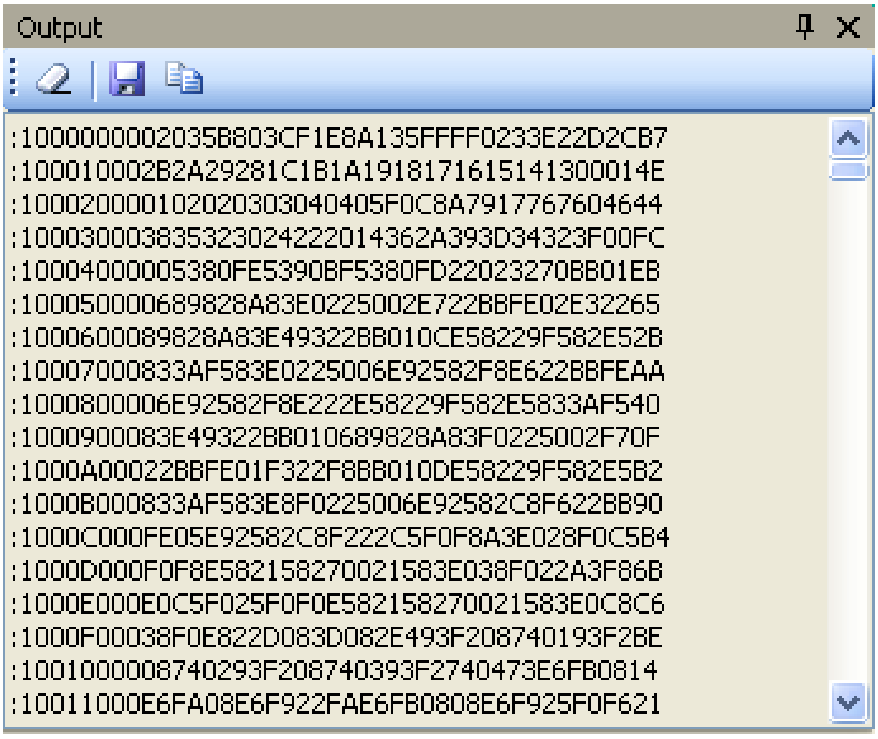
Select the Z-Wave Single Chip tab that match the chip connected to the programming socket. In our example ZW020x. Click on the **Read** button and a message will appear:



Please wait a moment until reading the Flash.

Figure 26. Flash reading

When the Flash content has been read, the output is shown in the **Output** view:



```
:1000000002035B803CF1E8A135FFFF0233E22D2CB7
:100010002B2A29281C1B1A1918171615141300014E
:100020000102020303040405F0C8A7917767604644
:100030003835323024222014362A393D34323F00FC
:1000400000005380FE5390BF5380FD22023270BB01EB
:100050000689828A83E0225002E722BBFE02E32265
:1000600089828A83E49322BB010CE58229F582E52B
:10007000833AF583E0225006E92582F8E622BBFEAA
:1000800006E92582F8E222E58229F582E5833AF540
:1000900083E49322BB010689828A83F0225002F70F
:1000A00022BBFE01F322F8BB010DE58229F582E5B2
:1000B000833AF583E8F0225006E92582C8F622BB90
:1000C000FE05E92582C8F222C5F0F8A3E028F0C5B4
:1000D000F0F8E582158270021583E038F022A3F86B
:1000E000E0C5F025F0F0E582158270021583E0C8C6
:1000F00038F0E822D083D082E493F208740193F2BE
:1001000008740293F208740393F2740473E6FB0814
:10011000E6FA08E6F922FAE6FB0808E6F925F0F621
```

Figure 27. Output of a HEX file reading

Now, the output can be saved to a file. Click on the **Save** icon and a dialog window will appear:

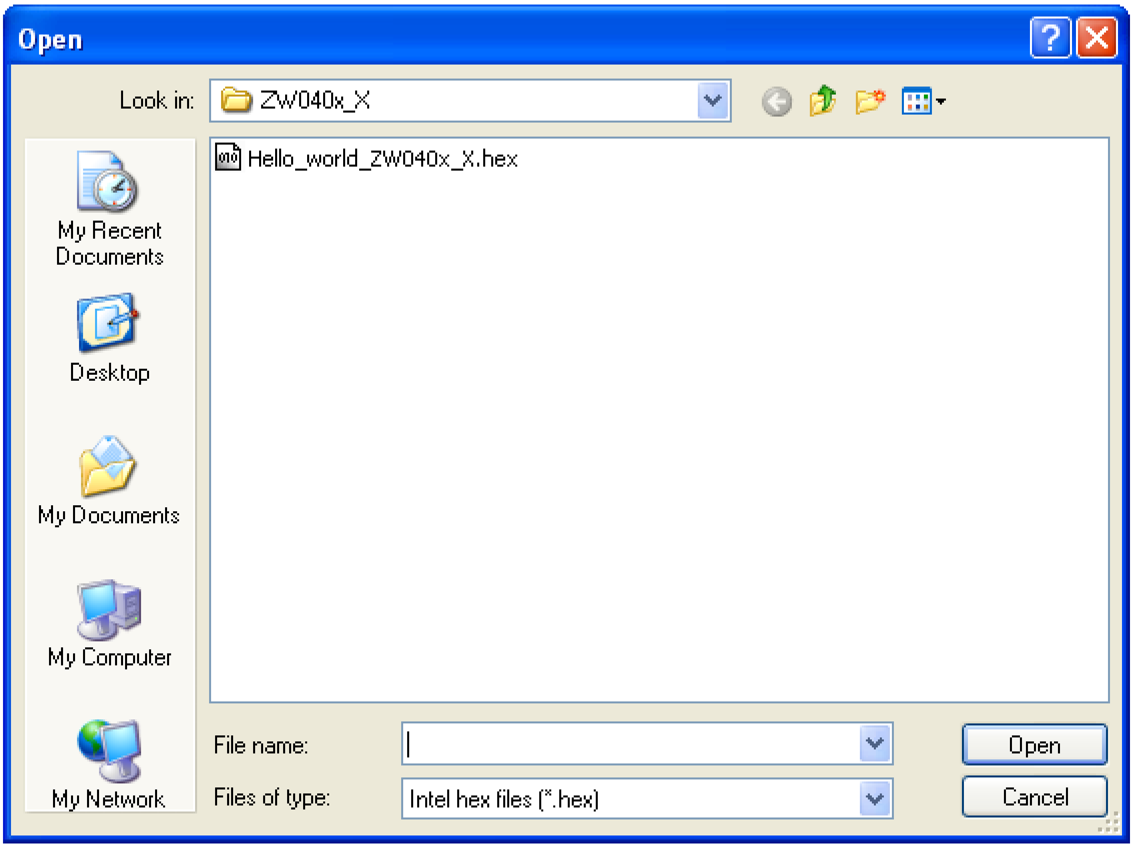


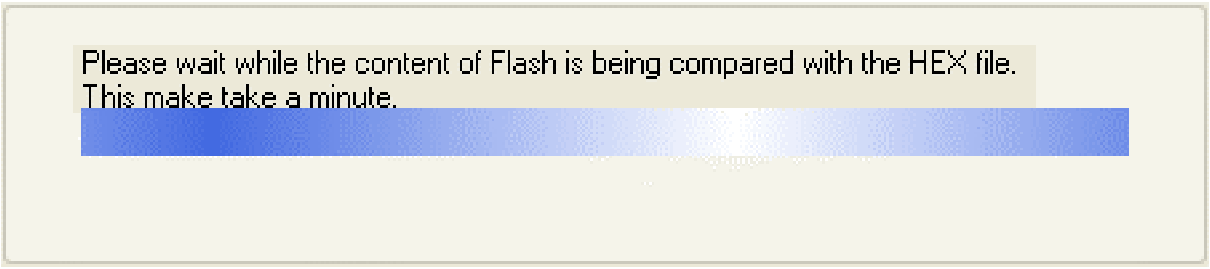
Figure 28. The HEX file name

Click on the **Save** button and the HEX file content will be saved to the designated location.

5.3.4 Compare Flash\OTP content with a HEX file.

Z-Wave Programmer can be used to compare the content of a Z-Wave Single Chip with a hex file as follows:

Click on the open HEX file button and browse to location of the HEX file in question. When a HEX file is selected, click on the **Compare** button. The process will start and the popup message will show the process progress.



Please wait while the content of Flash is being compared with the HEX file.
This make take a minute.

Figure 29. Progress of a HEX file comparison

After comparison, the appropriate info message will appear.

5.3.5 Calibration and Program (ZW040x only)

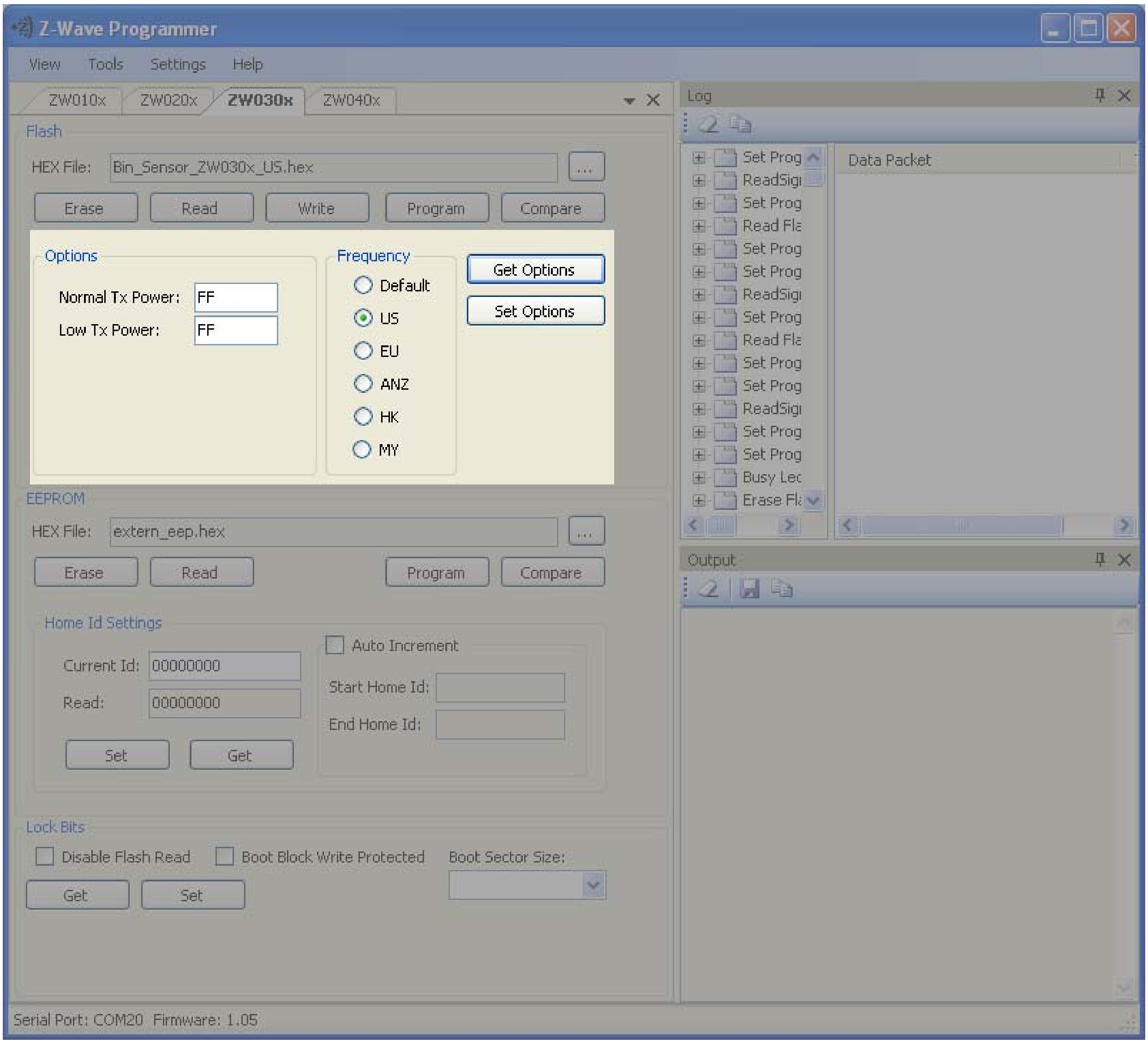
This option is possible if the hardware calibration tool is connected to the Z-Wave Programmer. The Calibration HEX file must be specified as well under Main Menu > Settings > Other > Calibration HEX.

The chip is calibrated before programming if this option is selected.

5.3.6 Writing RF flash settings[1]

The RF parameters can be changed directly on the ZW0102/ZW0201/ZW0301 Single Chips without changing the firmware. This leads to a faster approach when trying out different settings in the HW prototype phase.

The Z-Wave Programmer can write or read the RF flash settings of the whole range of Z-Wave single chips. To write the RF settings simply change the desired value then click on the ‘**Set Options’** button.



HEX File: Bin_Sensor_ZW030x_US.hex

Normal Tx Power: FF

Low Tx Power: FF

Frequency: ○ Default ● US ○ EU ○ ANZ ○ HK ○ MY

HEX File: extern_eep.hex

Current Id: 00000000

Read: 00000000

□ Auto Increment

Start Home Id:

End Home Id:

□ Disable Flash Read □ Boot Block Write Protected Boot Sector Size:

```
Set Prog
ReadSigı
Set Prog
Read Fla
Set Prog
Set Prog
ReadSigı
Set Prog
Read Fla
Set Prog
Set Prog
ReadSigı
Set Prog
Set Prog
Busy Lec
Erase Fla
```

Serial Port: COM20 Firmware: 1.05

Figure 30. Frequency options

To read the RF settings, click on the ‘**Get Options**’ button and all the RF setting will be read from the flash and shown in the appropriate fields.

[1] Changing the RF flash settings in ZW0102 cannot be done directly; the entire flash has to be programmed.

5.3.6.1 Editing the ZW0102 RF-settings

The “**Normal Tx Power**” can easily be changed in case it has to be higher/lower than anticipated during the HW prototype phase.

The numbers written into the entries are control values. In the case of TX power, the number controls the gain of a PA, and it is thus impossible to predict the resulting transmission power, because that will depend on the environment in which the Z-Wave Module is mounted. Mounting no filters between the RF_OUT on the Z-Wave Module and the antenna a number of 80_{hex} (80 hexadecimal letters, 128 decimal) as normal TX Power equal app. to 0dBm delivered from the ZW0102 to the antenna.

The numbers written in the power control registers “**Normal Tx Power**” and “**Low Tx Power**” have to be written in hexadecimal form, e.g. A0, or B5, or 1C. The PA consists of two parallel PA’s, each controlled by a nibble of the power register. Writing F0 is thus full power to the strongest PA and lowest possible power to the weakest PA of the two PA’s. Due to implementation details, FF is *not* a valid power-setting. Because the PA is implemented as two separate PA’s, one might get problems with harmonics when using both PA’s at the same time. The power might be higher when using the setting F1, but the harmonics are much lower when using F0.

Generally, making the number higher will increase the transmission power, lowering the number will decrease the transmission power. There is no linear relationship between the settings of the PA’s.

The “**Low Tx Power**” entry controls the transmission power in situations where it is natural that a Controller and a Slave are close to each other, e.g. when a Slave is added to, or removed from, a network.

The table below shows a typical mapping between RF-power radiated form the PA to the antenna, assuming no filters or other attenuation between the antenna and the PA. Notice that only a fraction of PA settings are shown as an example:

Table 1. ZW0102 PA Settings.

Power setting	**PA power**
Hexadecimal value	Valid for both normal and low power [dBm]
10	-17
30	-8
50	-3
70	-0.5
90	1.2
B0	2.8
D0	4
F0	5
1F	-4
3F	-1.3
5F	0.5
BF	4.3
DF	5

A precise mapping between radiated RF power from the antenna and the PA setting requires a measurement on the specific product.

The numbers written into the Match entries controls a capacitor match array connected to the RF_OUT pin on the ZW0102. The incremental step-size is 0.4pF, that is, increasing the TX Match with one, increases the capacitor value with 0.4pF. The same applies for the RX Match. The TX match register is used during transmission, where as the RX match register is used during reception of data.

Notice that it is not possible to use the '**Write Options'** button for the 100 Series Single Chip. The whole firmware must be downloaded to update the RF settings.

5.3.6.2 Editing the ZW0201/ZW0301 RF-settings

The "**Normal Tx Power**" can easily be changed in case it has to be higher/lower than anticipated during the prototype phase.

The bias of the PA, and thus the transmitted power, is controlled by the setting the register values from the table below into the "**Normal Tx Power** "and "**Low Tx Power"** fields. Both the current consumption of the ZW0201ZW0301 chip and the delivered power to the antenna are influenced by the configured register values. Selecting a value from the **High Power** column consumes much more current than selecting one from the **Low Power** column, but more power can be delivered to the antenna using the **High Power** setting.

The "**Low TX Power"** entry controls the transmission power in situations where it is natural that a Controller and a Slave are close to each other, e.g. when a Slave is added to, or removed from, a network.

The table below shows the **only** valid mappings between RF-power radiated from the PA to the antenna, assuming no filters or other attenuation between the antenna and the PA:

Table 2. ZW0201/ZW0301 PA Settings.

High Power from PA		**Low Power from PA**	
High power [dBm]	Register setting (hex)	Low power [dBm]	Register setting (hex)
		-19	0x14
		-17	0x15
-7	0x26	-15	0x16
-5	0x27	-13	0x17
-3	0x28	-11	0x18
-1	0x29	-9	0x19
0.5	0x2A	-7	0x1A
		-5	0x1B

A precise mapping between radiated RF power from the antenna and the PA setting requires a measurement on the specific product.

5.3.6.3 ZW040x RF settings

ZW040x RF settings cannot be edited, but can be read out from the device, under **OTP Memory > Options > Get Options**.

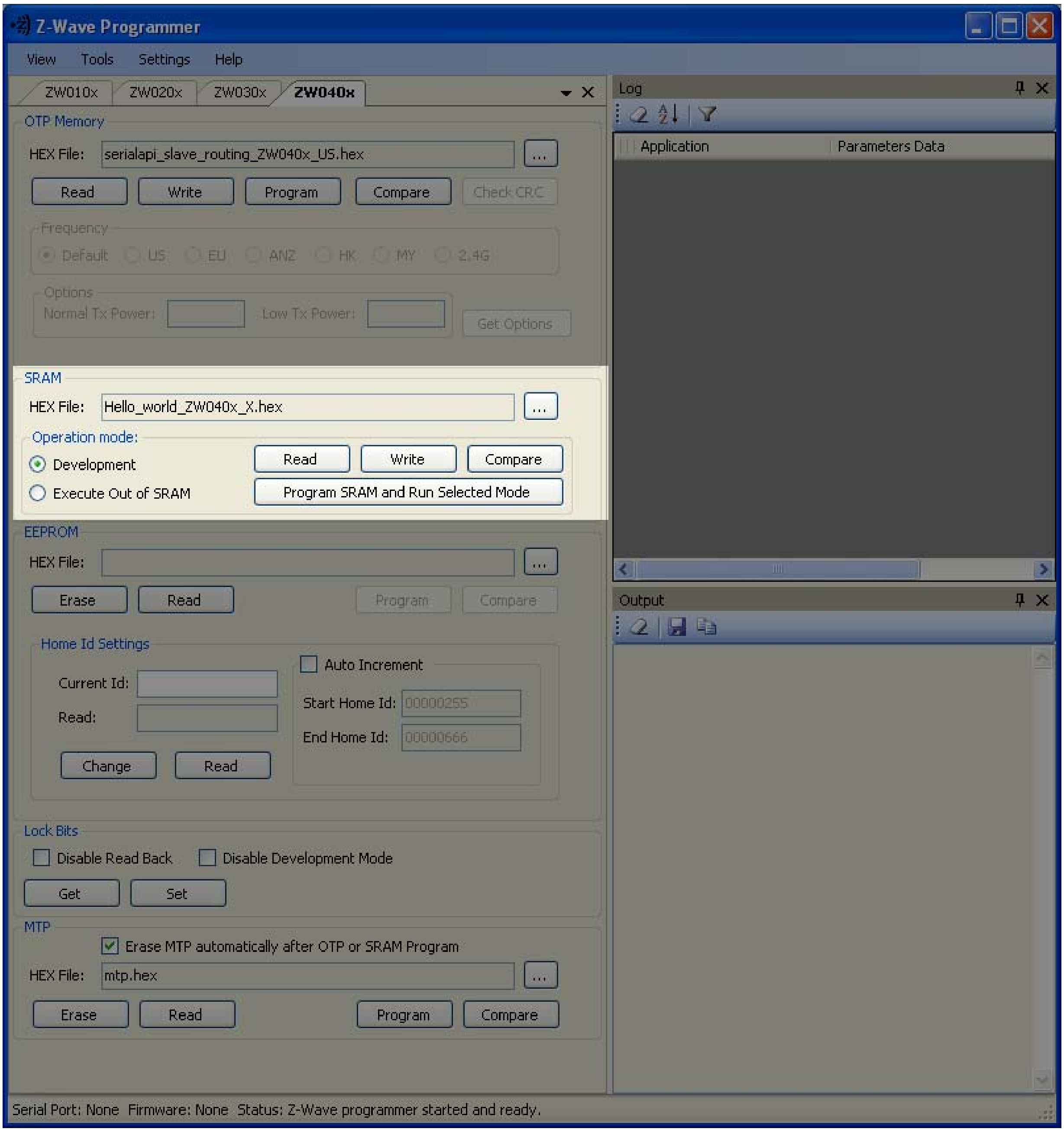
Table 3. RF normal and low power settings ZW040x

SW Power settings	**RFPOW Register**	**Output Power (dBm)**
	0x30	4.4dBm
Normal	0x20	3dBm

	0x1E	2dBm
Normal -2dB	0x18	0.8dBm
	0x14	-0.3dBm
Normal -4dB	0x10	-2dBm
	0x0E	-3.4dBm
Normal -6dB	0x0C	-4.7dBm
Normal -8dB	0x0A	-6.1dBm
Normal -10dB	0x08	-8dBm
Normal -12dB	0x06	-10.3dBm
Normal -14dB	0x05	-11.4dBm
Normal -16dB	0x04	-13.8dBm
Normal -18dB	0x03	-15.2dBm
Normal -20dB	0x02	-19.5dBm
Normal -22dB	0x01	-23dBm

5.4 **SRAM handling** (ZW040x only)

This section applies only to ZW040x Single Chips.



HEX File: serialapi_slave_routing_ZW040x_US.hex

HEX File: Hello_world_ZW040x_X.hex

Operation mode:

● Development

○ Execute Out of SRAM

□ Auto Increment

Start Home Id: 00000255

End Home Id: 00000666

□ Disable Read Back □ Disable Development Mode

☑ Erase MTP automatically after OTP or SRAM Program

HEX File: mtp.hex

Serial Port: None Firmware: None Status: Z-Wave programmer started and ready.

Figure 31. SRAM

The 400 series Z-Wave Signal Chip has two SRAM blocks that can be accessed from the SPI interface.

The 4 kbyte SRAM is used as code memory in the “Execute out of SRAM” mode.

A 12 kbyte SRAM is used in development mode as code memory.

5.4.1 SRAM operation modes

SRAM can be operated in two modes:

- Development;
- Execute out of SRAM

5.4.1.1 Development SRAM operation mode

In Development mode, the 12 kbyte SRAM overlays the upper 12 kbytes of the OTP memory. This enables a way of running application SW that is under development without having to program the OTP.

5.4.1.2 Execute Out of SRAM operation mode

In this mode, the chip operates the 4 Kbytes SRAM.

5.4.2 SRAM operation functions

5.4.2.1 Read

The content of SRAM can be read by activating the **Read** button. The content of the SRAM will be read and displayed in the **Output** view.

5.4.2.2 Write

Select a HEX file to be written to SRAM in the file location bar, and activate the **Write** button. The file will be written to the SRAM.

5.4.2.3 Compare

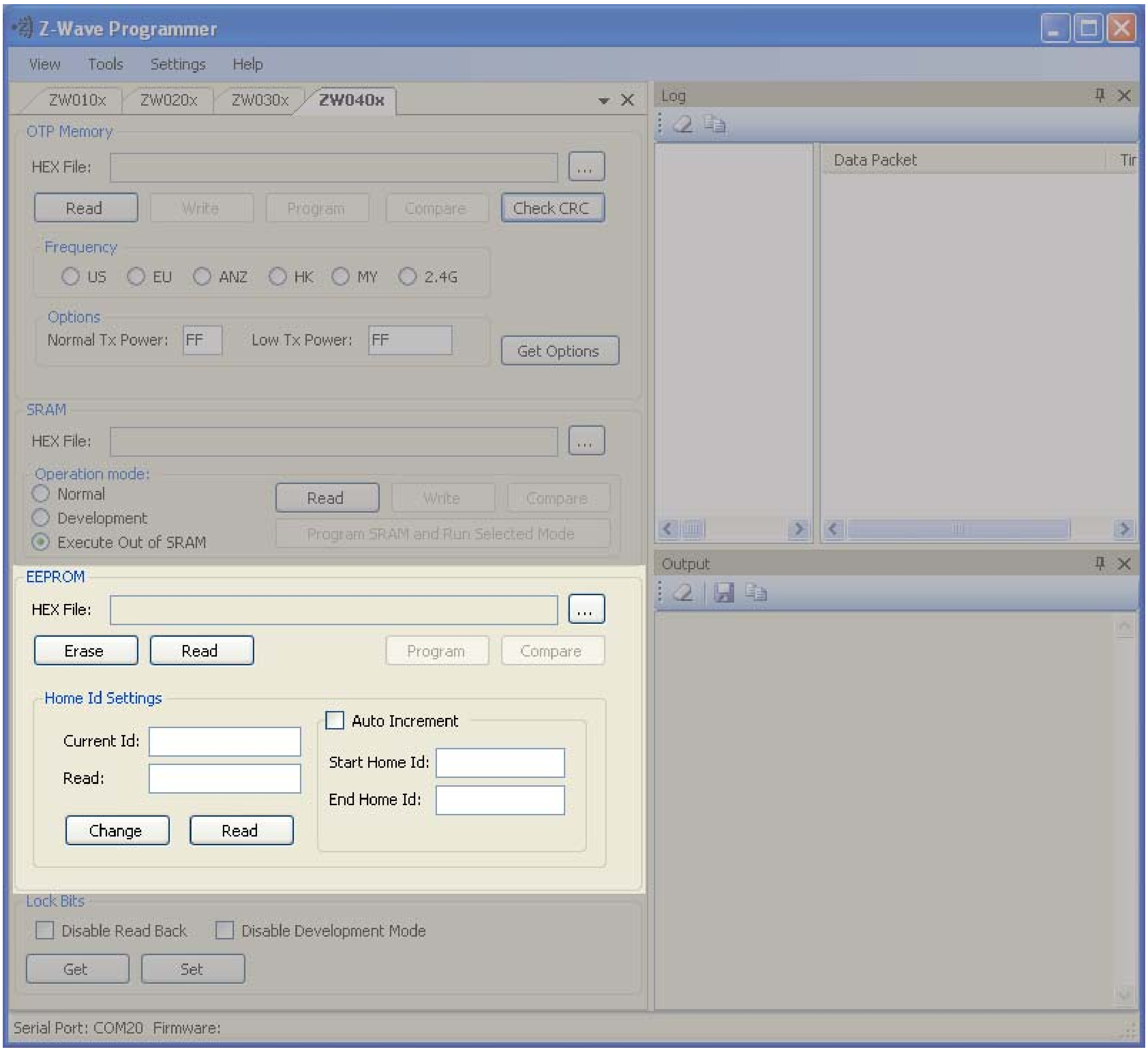
Select a HEX file to compare in the file location bar, and activate the **Compare** button. The SRAM content will be compared with the selected file, a procedure alike Flash/OTP comparison.

5.4.2.4 Program SRAM and Run Selected Mode

Select a HEX file to be programmed to SRAM in the file location bar, and activate the **Program SRAM and Run Selected Mode** button. The file will be written to the SRAM with verification, and the selected Operation Mode will be run.

5.5 External EEPROM handling

The Z-Wave Programmer can only handle a 32KB EEPROM as specified for the Z-Wave modules.



Normal Tx Power: FF

Low Tx Power: FF

● Execute Out of SRAM

Serial Port: COM20 Firmware:

Figure 32. EEPROM

5.5.1 Writing a EEPROM file to the EEPROM[2]

Select a HEX file to be written to EEPROM in the file location bar, and activate the **Program** button. The file will be written to the EEPROM. This process will take a few moments to complete.

If any error occurs during programming, a failure message will appear in a pop-up window.

5.5.2 Clear the EEPROM content

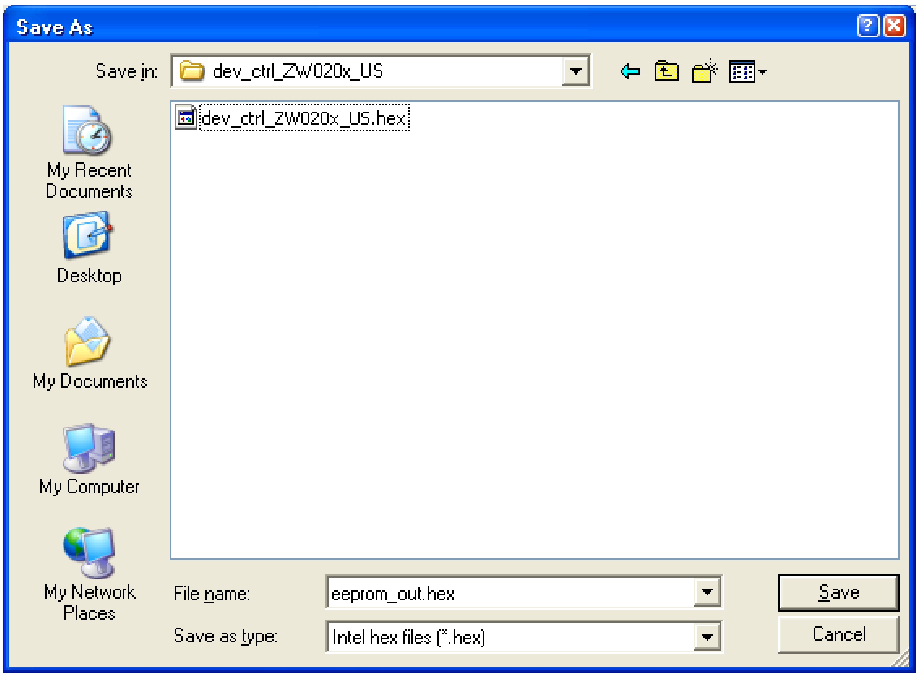
To clear the entire EEPROM incl. the Home ID, simply click on the '**Erase'** button in the EEPROM group box. The EEPROM clearing process will start and take a few moments to complete.

5.5.3 Reading the EEPROM content

The Z-Wave Programmer can read the content of an EEPROM mounted on a Z-Wave module and save it to a file. To read the content of the EEPROM, do the following:

Click on the '**Read**' button, and the content of the EEPROM will be read. This process will take a few moments. When finished, the content will be displayed in the **Output** view.

Then it is possible to save the obtained content to a HEX file. Click on the '**Save**' icon in the **Output** view, and the '**Save As**' dialog will appear. Here the target folder for the EEPROM content must be chosen.



Save in: dev_ctrl_ZW020x_US

dev_ctrl_ZW020x_US.hex

File name: eeprom_out.hex

Save as type: Intel hex files (*.hex)

Figure 33. The EEPROM content file name

Click on the 'Save' button, and the file will be saved.

[2] The ZWaveProgrammer cannot write to an EEPROM mounted on ZM1206 modules.

5.5.4 Comparing the EEPROM content

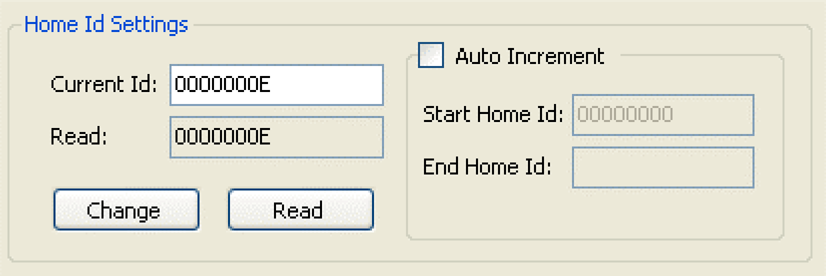
Z-Wave Programmer can be used to compare the content of an EEPROM with a hex file as follow:

Click on the open '**HEX file**' button and browse to the location of the EEPROM file in question. After the EEPROM file was selected, click on the 'Compare' button. The process will start and the status pop-up window will show the process progress.

When finished, the appropriate pop-up window will appear with information on success or failure.

5.5.5 Home ID manipulation[3]

The Z-Wave Programmer can be used to change or read a Z-Wave module's home ID. To change the home ID, write the desired home ID in the '**Current ID**' edit box then click on the '**Change**' button. The new home ID will be written to the EEPROM and the home ID in the EEPROM will be read and displayed on the textbox named '**Read'**. The home ID stored in the EEPROM can be read by the Z-Wave Programmer, simply click on the '**Read**' button and the home ID will be shown on the '**Read**' textbox.



Home Id Settings

Current Id: 0000000E

Read: 0000000E

□ Auto Increment

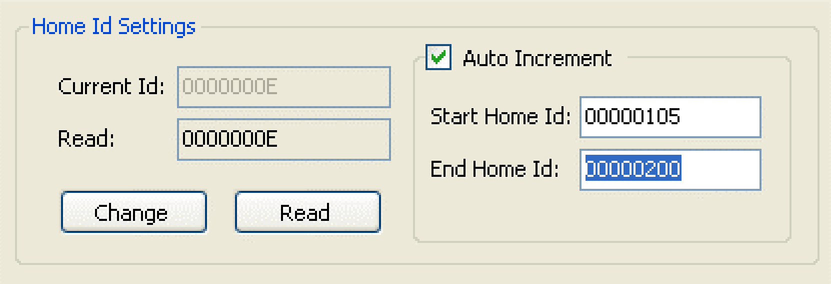
Start Home Id: 00000000

End Home Id:

Figure 34. The home ID after change operation done

The Z-Wave Programmer can also be used to setup a range of home IDs so that each time the '**Change HomeID'** button is click a new home ID from the range will be used. To use this feature, check the '**Auto increment**' checkbox and set a start/end home ID in the '**Start Home ID**' and '**End Home ID**' edit boxes respectively.

Click on the '**Change**' button, and the next available home ID in the range will be shown in the '**Current ID**' textbox and written to the EEPROM as shown in Figure 35.



Home Id Settings

Current Id: 0000000E

Read: 0000000E

☑ Auto Increment

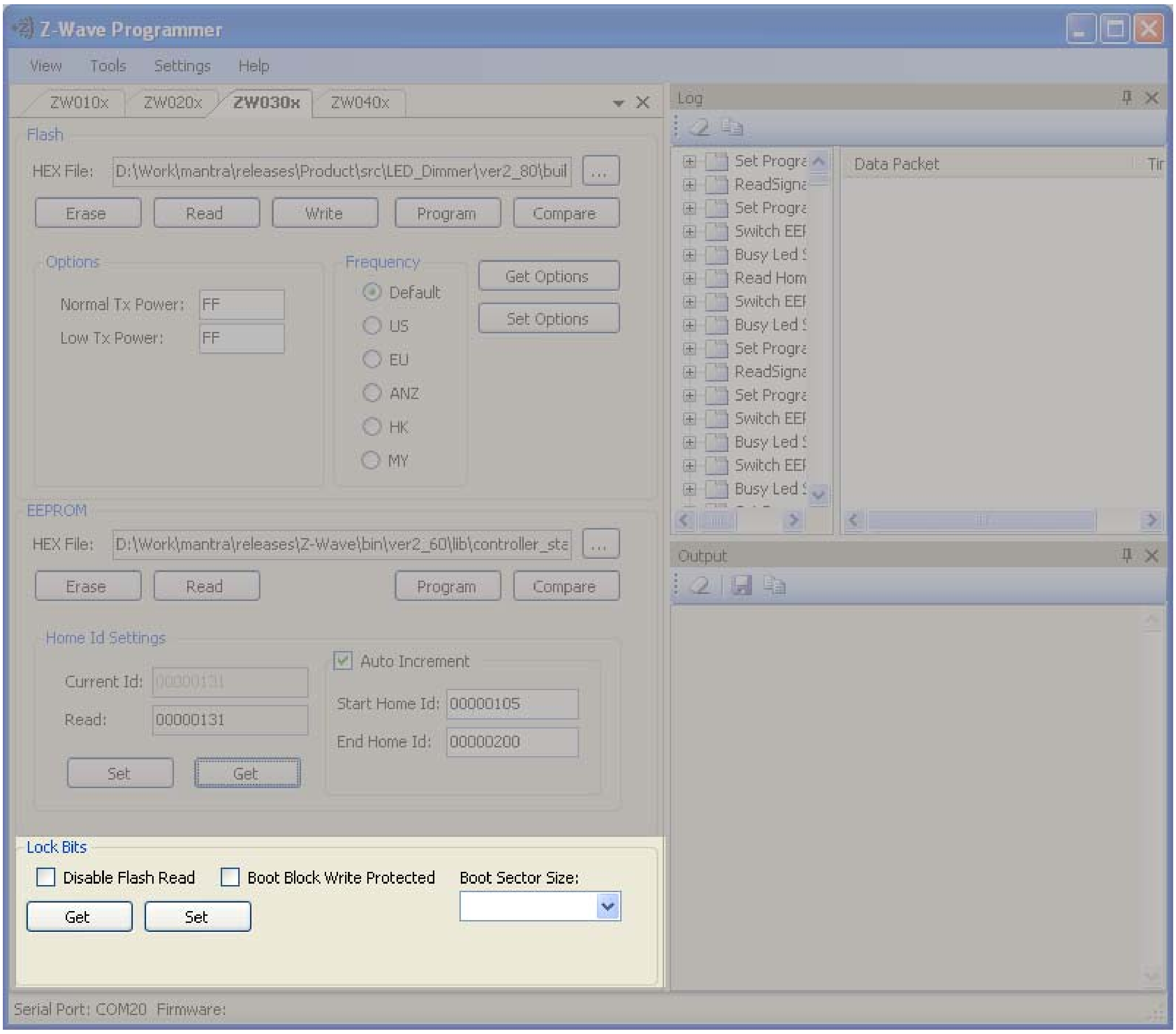
Start Home Id: 00000105

End Home Id: 00000200

Figure 35. Auto increment home ID in action

[3] The ZWaveProgrammer cannot write to an EEPROM mounted on ZM1206 modules

5.6 Lock bits



Flash HEX File: D:\Work\mantra\releases\Product\src\LED_Dimmer\ver2_80\buil

Normal Tx Power: FF

Low Tx Power: FF

Frequency: ● Default ○ US ○ EU ○ ANZ ○ HK ○ MY

EEPROM HEX File: D:\Work\mantra\releases\Z-Wave\bin\ver2_60\lib\controller_sta

Current Id: 00000131

Read: 00000131

☑ Auto Increment

Start Home Id: 00000105

End Home Id: 00000200

□ Disable Flash Read □ Boot Block Write Protected Boot Sector Size:

```
Set Progra
ReadSigna
Set Progra
Switch EEF
Busy Led S
Read Hom
Switch EEF
Busy Led S
Set Progra
ReadSigna
Set Progra
Switch EEF
Busy Led S
Switch EEF
Busy Led S
```

Serial Port: COM20 Firmware:

Figure 36. Lock bits

The function of the Lock bits is to:

- Enable read-back protection of the entire flash;
- Enable protection of page 0 against inadvertent writing;
- Enable and set the size of a boot sector (a part of the flash) which is secured against inadvertent writing.

5.6.1 Obtain Lock Bits status

To obtain the status of Lock Bits, press the ‘**Get**’ button. The current values of the Lock Bits will be read and displayed.

5.6.2 Disable Flash Read

To enable read-back protection of the entire flash, check the ‘**Disable Flash Read**’ checkbox and press the ‘**Set**’ button.

5.6.3 Boot Block Write protected

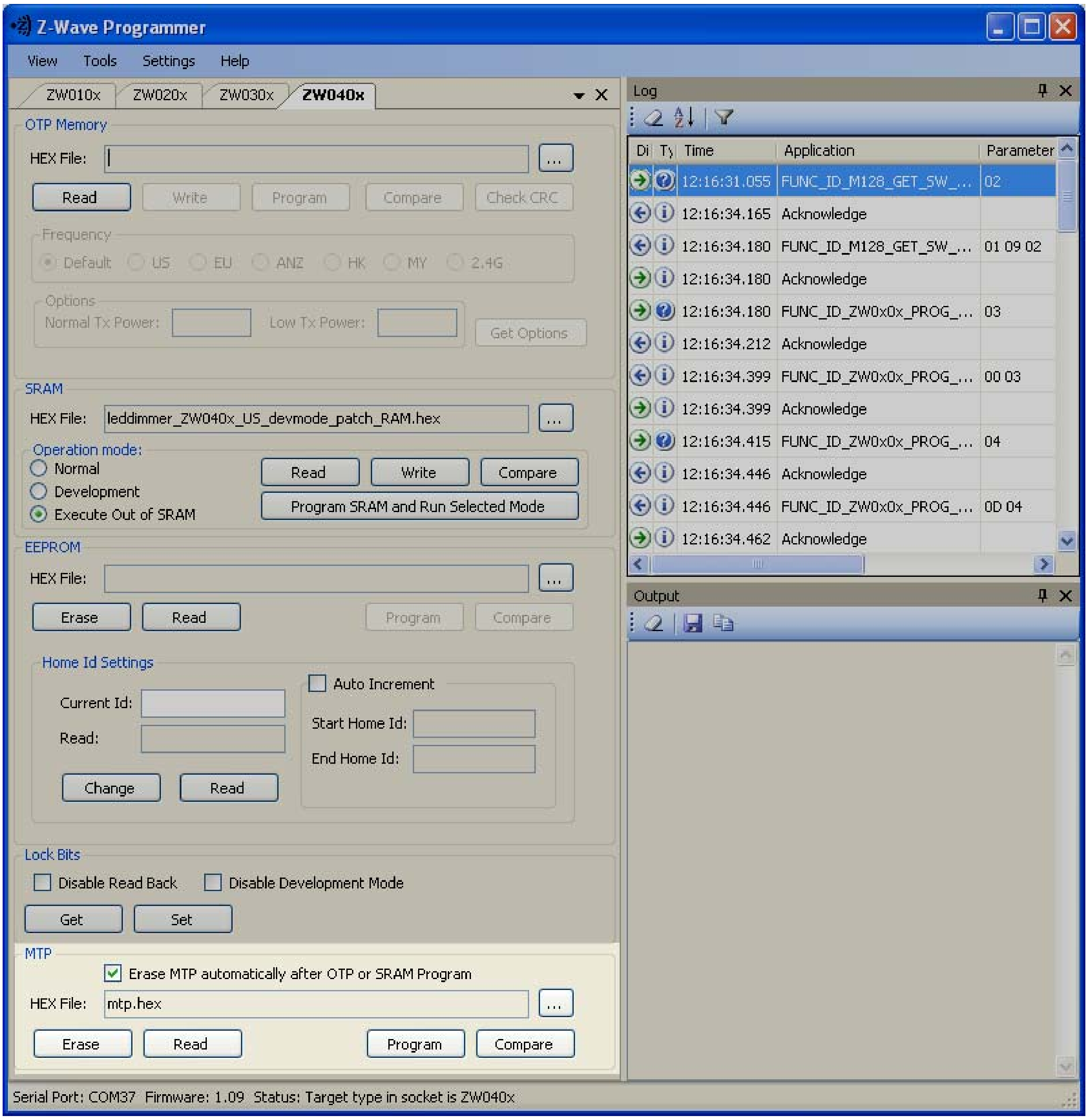
To enable protection of page 0 against inadvertent writing, check the ‘**Boot Block Write protected**’ checkbox and press the ‘**Set**’ button.

5.6.4 Boot Sector Size

To enable and set the size of a boot sector (a part of the flash) which is secured against inadvertent writing, select the desired boot sector size from the dropdown list and press the ‘**Set**’ button.

For more information on the Lock Bits programming, please see Reference [3].

5.7 MTP



SRAM HEX File: leddimmer_ZW040x_US_devmode_patch_RAM.hex

Operation mode:
○ Normal
○ Development
● Execute Out of SRAM

MTP
☑ Erase MTP automatically after OTP or SRAM Program
HEX File: mtp.hex

Di	Ty	Time	Application	Parameter
→	?	12:16:31.055	FUNC_ID_M128_GET_SW_...	02
←	i	12:16:34.165	Acknowledge	
←	i	12:16:34.180	FUNC_ID_M128_GET_SW_...	01 09 02
→	i	12:16:34.180	Acknowledge	
→	?	12:16:34.180	FUNC_ID_ZW0x0x_PROG_...	03
←	i	12:16:34.212	Acknowledge	
←	i	12:16:34.399	FUNC_ID_ZW0x0x_PROG_...	00 03
→	i	12:16:34.399	Acknowledge	
→	?	12:16:34.415	FUNC_ID_ZW0x0x_PROG_...	04
←	i	12:16:34.446	Acknowledge	
←	i	12:16:34.446	FUNC_ID_ZW0x0x_PROG_...	0D 04
→	i	12:16:34.462	Acknowledge	

Serial Port: COM37 Firmware: 1.09 Status: Target type in socket is ZW040x

Figure 37. MTP

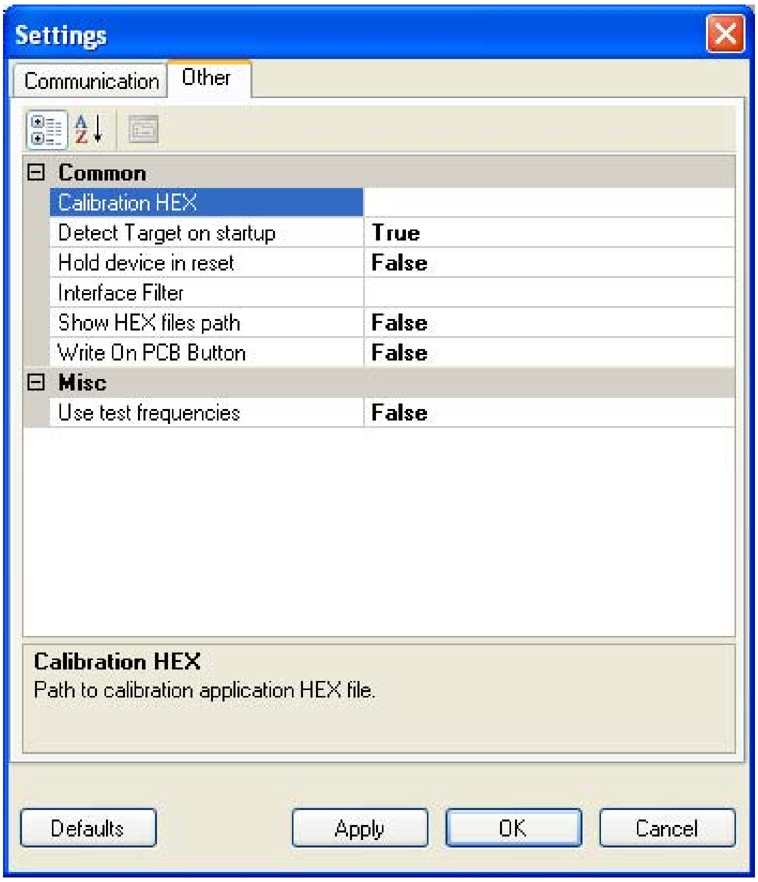
The 64-byte MTP (Many Time writeable Memory) was introduced in the 400 Series Single Chips with the purpose to minimize the size of the non-volatile memory on the chip for a routing slave without changing the fundamental way it operates in the network. It is capable of storing the node’s Home ID (only for routing slaves), Node ID, SUC data, Associations data.

The “Erase MTP automatically after OTP or SRAM Program” checkbox is active by default, so the Programmer erases MTP before programming in normal mode, development mode and ’execution out of SRAM’ mode. It is possible to deselect the MTP erase step.

The other controls are as follows:

- **Erase** - to erase the MTP content;
- **Read** – to read the MTP content with further possibility to save it to a specified file.
- **Program** – to write the target file to MTP;
- **Compare** – to compare the MTP content with the file specified.

5.8 Miscellaneous features



Common	
Calibration HEX	
Detect Target on startup	True
Hold device in reset	False
Interface Filter	
Show HEX files path	False
Write On PCB Button	False
Misc	
Use test frequencies	False

Calibration HEX
Path to calibration application HEX file.

Figure 38. Other settings

5.8.1 Reset Z-Wave Module

Main Menu > Tools > Reset Z-Wave Module

This function send reset signal to the module.

5.8.2 Calibration HEX

Main menu > Settings > Other

Specify the path to the Calibration HEX file which is used in Calibration process.

5.8.3 Detect Target on startup

Main menu > Settings > Other

Switch ‘Detect Target on startup’ to **True.** The Programmer will detect the target on startup.

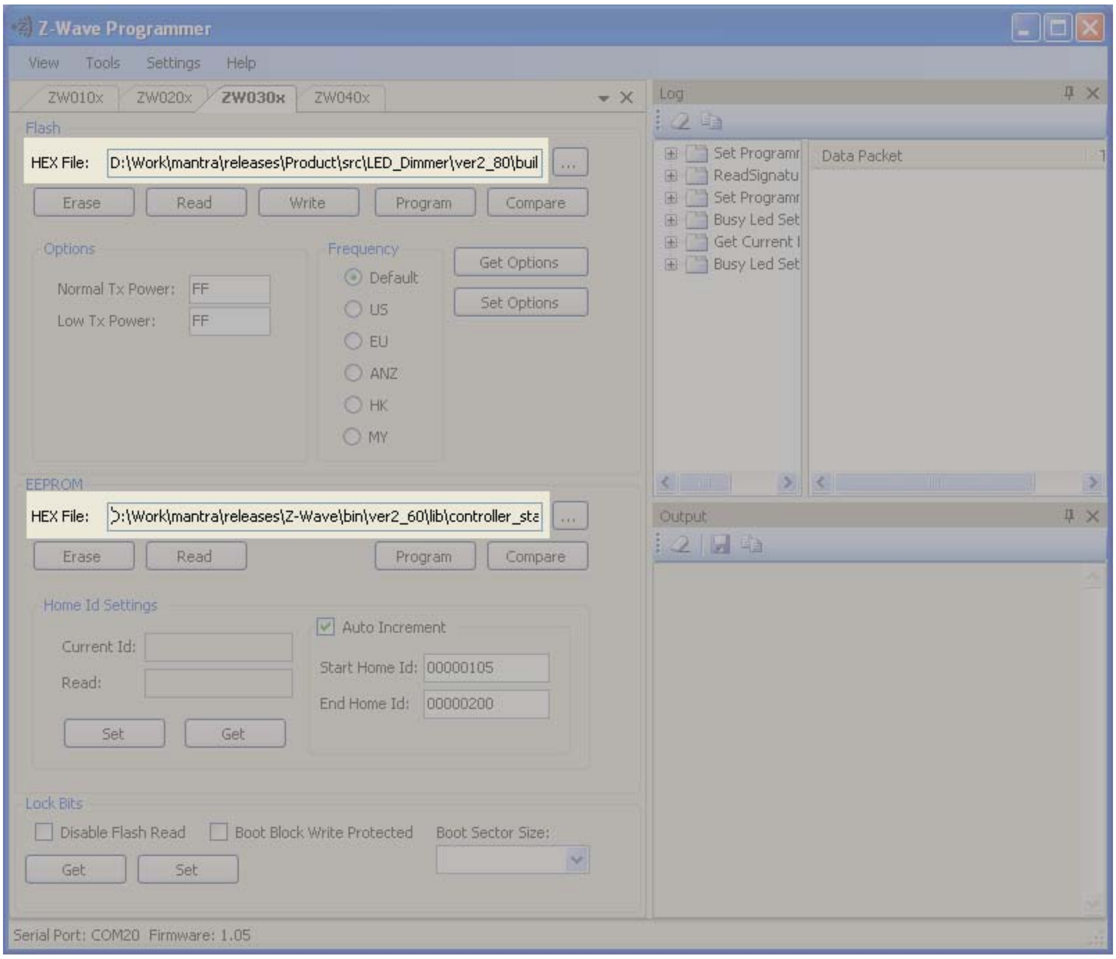
5.8.4 Hold device in reset

Main menu > Settings > Other

If set to **True**, reset signal will be sent to the programmed until the power is cycled.

5.8.5 Show HEX file paths

When this option is selected, the Z-Wave Programmer will show the full path of the flash and the EEPROM HEX files as shown in Figure 39:



Flash HEX File: D:\Work\mantra\releases\Product\src\LED_Dimmer\ver2_80\buil

Normal Tx Power: FF

Low Tx Power: FF

● Default ○ US ○ EU ○ ANZ ○ HK ○ MY

EEPROM HEX File: Ɔ:\Work\mantra\releases\Z-Wave\bin\ver2_60\lib\controller_sta

☑ Auto Increment

Start Home Id: 00000105

End Home Id: 00000200

□ Disable Flash Read □ Boot Block Write Protected Boot Sector Size:

Serial Port: COM20 Firmware: 1.05

Figure 39. Z-Wave Programmer with Show HEX files paths enabled

When the option is not selected, the Z-Wave Programmer will only show the HEX files names.

5.8.6 Interface Filter

This option serves to narrow the search of the required PC interface port for the Z-Wave Programmer to work with. Any keyword, e.g. ‘UART’ can be typed in the text field working as a filter, as shown in Figure 38.

5.8.7 Write on PCB Button

If this option is enabled, the Flash is programmed with the specified HEX file when S1 button is pressed on the Programmer board.

5.8.8 Use test frequencies

If this option is enabled, the **Set Options** function will set a test frequency for the device according to the radiobutton frequency selector.

5.9 Keyboard shortcuts

The GUI is extended with a number of shortcut keys to provide an easier and usually quicker method of navigating the Z-Wave Programmer. The table below lists the available shortcut keys.

Table 4. Z-Wave Programmer shortcut keys.

Z-Wave Programmer features	Shortcut Keys
Detect target	Ctrl + D
Select ZW010x target	F1
Select ZW020x target	F2
Select ZW030x target	F3
Select ZW040x target	F4
Open a flash HEX file	Ctrl + O
Program flash (write then verify)	Ctrl + P
Write flash (Write only)	Ctrl + W
Erase flash	Ctrl + E
Read flash	Ctrl + R
Compare flash content with a HEX file	Ctrl + M
Check CRC (ZW040x only)	Ctrl+C
Write RF settings (except ZW040x)	SHIFT + W
Read RF settings (except ZW040x)	SHIFT + R
Open an EEPROM HEX file	Alt + O
Program EEPROM	Alt + P
Clear EEPROM	Alt + E
Read EEPROM	Alt + R
Compare EEPROM content with a HEX file	Alt + M
Change home ID in EEPROM	Alt + F1
Read home ID from EEPROM	Alt + F2
Read SRAM (ZW040x only)	Ctrl + Alt + R
Write SRAM (ZW040x only)	Ctrl + Alt + W
Compare SRAM (ZW040x only)	Ctrl + Alt + C
Program SRAM and run selected mode (ZW040x only)	Ctrl + Alt + P
Exit	Alt+X

The shortcut keys can also be obtained directly from the GUI. Simply mouse hover over a control to obtain a tool tip showing the shortcut key to this particular control.

5.10 Z-Wave Programmer console interface

In addition to the Z-Wave Programmer GUI, the Z-Wave Programmer has a console interface. The console interface is useful when using the Z-Wave Programmer from the command prompt

5.10.1 Console interface syntax

The Console interface has the following syntax:

ZWaveProgrammer.exe -c COMx [-t chipType] [-pf frequency] [-p filename] [-f filename] [-r filename] [-v filename] [-e] [-ls lockbitsbyte] [-lg] [-sr filename] [-sw filename] [-sc filename] [-swrd filename] [-sweo filename] [-mp filename] [-mp 0] [-mr filename] [-pe filename] [-pe 0] [-pr filename] [-ph HomeID]

-c - The comport to use in the format COMx: where x is number

-t - The Chip Type (ZW010x, ZW020x, ZW030x, ZW040x).

5.10.2 Flash/OTP Programming options:

-p - Erase chip (flash only), Program and verify pages that are not empty.

-f - Erase chip (flash only), Program chip

-r - Reads the flash/OTP contents and writes it to the file specified

-v - Verify chip against file supplied

-e - Erase chip (flash only)

-ls - Set lockbits.

-lg - Get lockbits.

-pf - Set the frequency (EU, US, ANZ, HK, MY, 866 (866.42MHz EU_tf), 870 (870.42MHz), 906 (906.42MHz US_tf), 910 (910.42MHz)) (used after -p or -f).

-ro – Read options.

-so – Set options.

5.10.3 ZW040x Programming options:

-sr - Read the SRAM contents and writes it to the file specified.

-sw - Write the SRAM contents with the file specified.

-sc - Compare the SRAM contents with the file specified.

-swrd - Write the SRAM contents with the file specified and run Development mode.

-sweo - Write the SRAM contents with the file specified

and Exec out of SRAM mode.

-sro - SRAM Read options.

-sso - SRAM Set options (options: [normal power ch0 byte] [low power ch0 byte] [normal power ch1 byte] [low power ch1 byte] [normal power ch2 byte] [low power ch2 byte]).

-mp - Programs the MTP memory with the file specified.

Zero fill empty areas.

-mp 0 - Erase entire MTP memory.

-mr - Reads MTP memory to file specified.

5.10.4 EEPROM Programming options :

-pe - Programs the external eeprom with the file specified.

Zero fill empty areas.

-pe 0 - Erase entire external eeprom.

-pr - Reads extern eeprom to file specified.

-ph - Set HomeID of external EEPROM with the supplied address: e.g. DEADBEEF.

Note: MTP memory is erased during programming OTP or SRAM.

Note: in Normal working mode of the ZW040x chip, any operation with MTP memory will leave ZW040x chip in the reset state. You need to cycle the power of the chip to switch its working mode to Normal and start execution of the embedded application from OTP.

5.10.5 Examples

Program the flash and set the frequency to US

```
ZWaveProgrammer –c com1 –pf US –p target.hex
```

Erase the flash

```
ZWaveProgrammer –c com1 –e
```

Read the ASIC flash content to the file read.hex

```
ZWaveProgrammer –c com1 -r read.hex
```

Compare the content of ASIC flash to the content of the hex file c.hex

```
ZWaveProgrammer –c com1 –v cp.hex
```

Write the content of the hex file eep.hex to the external EEPROM

```
ZWaveProgrammer –c com1 –pe eep.hex
```

Reads the contents of the EEPROM and write it to the file out.hex

```
ZWaveProgrammer –c com1 –pr out.hex
```

Change the home ID in the external EEPROM to 12345678

```
ZWaveProgrammer –c com1 –ph 12345678
```

5.10.6 Error codes

ZWaveProgrammer returns error code when used in console mode. This is required for proper functioning of the automated programming script.

If the requested operation has been completed successfully, the returned error code is zero. Otherwise, see the full list of error codes:

Error code	Explanation
1	Could not compare EEPROM content with the HEX file
2	Could not compare Flash content with the HEX file
3	Could not compare SRAM content with the HEX file
4	Could not detect the Z-Wave device
5	Could not erase the EEPROM content
6	Could not erase the Flash content
7	Could not get the firmware version
8	Could not initialize EEPROM
9	Could not read the Lock Bits
10	Could not set the Lock Bits. If you are trying to write lock bits not for the first time - erase device first. If the device is not erasable - replace it by a new one.
11	Could not read the application RF settings from Flash
12	Could not read the EEPROM content
13	Could not read the EEPROM options
14	Could not read the Flash content
15	Could not read the Flash options
16	Could not read the general options from Flash
17	Could not read the Home ID
18	Could not read the SRAM content
19	Cannot reset connected Z-Wave Module
20	Could not switch the device to BootLoader mode.
21	Could not switch the chip operation mode. Required operation mode may be disabled by a lock bit.
22	Could not switch the chip to Programming mode
23	Could not upgrade the firmware of the device
24	Could not write the application RF settings to Flash
25	Could not write content of the HEX file to EEPROM
26	Could not write content of the HEX file to Flash
27	Could not write the application RF settings to Flash
28	Could not write the general options to Flash
29	Could not write the Home ID
30	Could not write content of the HEX file to SRAM
31	Comparison of EEPROM content with the HEX file failed
32	Comparison of SRAM content with the HEX file failed

Error code	Explanation
33	The HEX file for EEPROM is not specified or specified file does not exist.
34	The HEX file for Flash is not specified or specified file does not exist.
35	Hex file not valid. Address out of range.
36	Hex file not valid
37	End value for Home ID must be greater than Start value
38	RF frequency was not selected
39	'Start Home Id' or 'End Home Id' was not specified
40	Undefined general option was encountered
41	Could not initialize MTP
42	Could not read the MTP content
43	Could not erase the MTP content
44	Could not program MTP
45	Could not compare MTP content with the HEX file
46	Programming of MTP failed
47	Comparison of MTP content with the HEX file failed
48	HEX file for MTP not specified
49	Reading the HEX file failed
251	Other errors
0	No error (operation has been successfully completed)

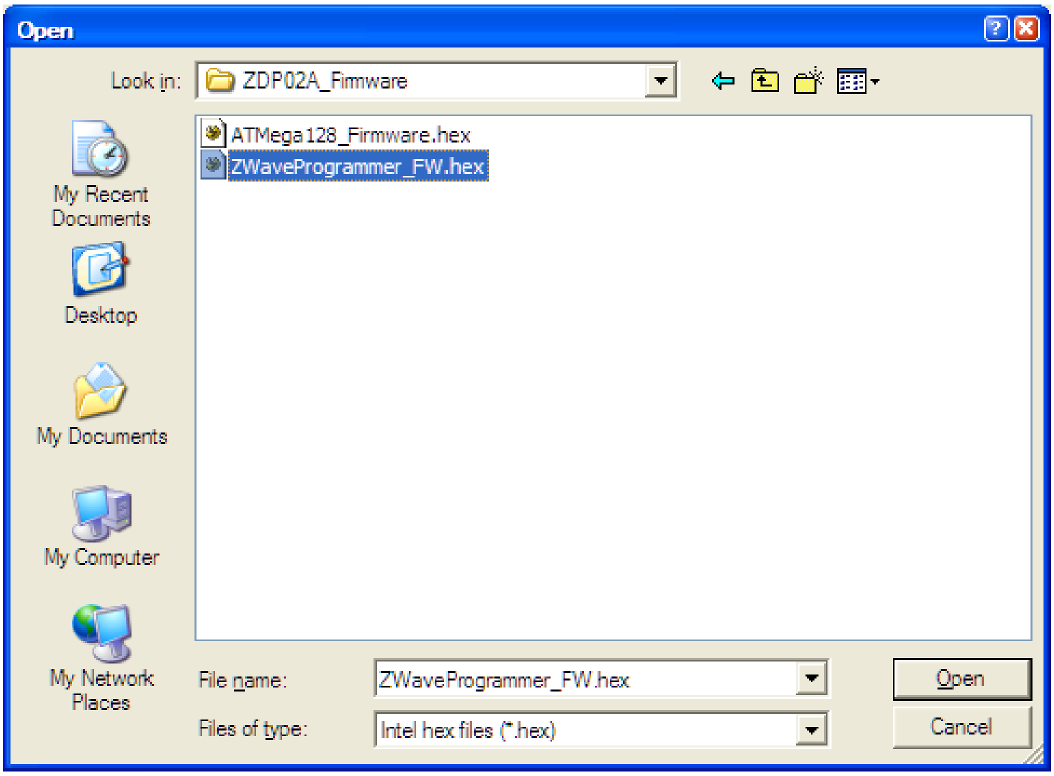
5.11 Z-Wave Programmer firmware update

5.11.1 Manual firmware upload

The Z-Wave Programmer firmware can be updated with a newer version directly from the Z-Wave Programmer GUI.

To open the firmware update dialog select **Tools > Upload Firmware...**

Select the Z-Wave Programmer firmware hex file and click 'Open':



ZDP02A_Firmware

ATMega128_Firmware.hex

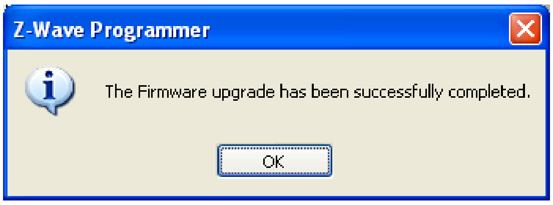
ZWaveProgrammer_FW.hex

File name: ZWaveProgrammer_FW.hex

Files of type: Intel hex files (*.hex)

Figure 40. Z-Wave Programmer firmware HEX file

If the upgrade was successful, the appropriate message will be displayed:



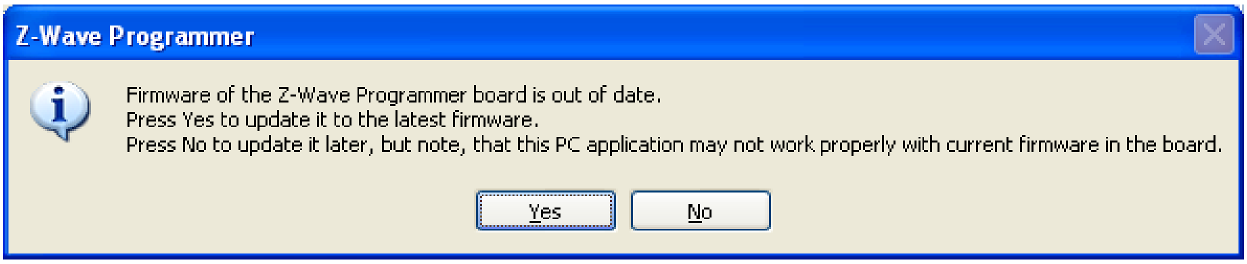
The Firmware upgrade has been successfully completed.

Figure 41. Upgrade operation is done.

You can read the firmware version in the Status Bar at the bottom of the Z-Wave Programmer window.

5.11.2 Automatic firmware version check and update

Z-Wave Programmer checks PCB firmware version during startup. If the firmware version in the Programmer board is older than the version known to Z-Wave Programmer application, the following message will pop up:



Z-Wave Programmer

Firmware of the Z-Wave Programmer board is out of date.
Press Yes to update it to the latest firmware.
Press No to update it later, but note, that this PC application may not work properly with current firmware in the board.

Figure 42. Firmware is out of date.

The user may choose to upgrade to the latest firmware version or to leave the current version running. However, the latter is not recommended.

5.12 Calibration

For ZW040x chips, it is possible to perform calibration. The purpose of calibration is to adjust the RF of the chip for stable communication avoiding RF deviations caused by the physical nature of the chip.

The calibration option is possible if a special hardware calibration tool is connected to the Z-Wave Programmer.

The Calibration HEX file must be specified under **Main menu > Settings > Other**. The calibration tool reads the values from it.

Each time when a ZW040x chip is going to be programmed, the Z-Wave Programmer warns the user that the calibration is not performed.

It is possible to program the chip with preliminary calibration through usage of the **Calibration and Program** button.

It is possible to only calibrate the chip: **Tools > Calibrate**.

It is possible to read out the calibration value of a ZW040x chip under **OTP Memory > Options > Get Options**.

REFERENCES

[1] Zensys, INS10247, Instruction, Z-Wave ZW0102/ZW0201/ZW0301 Application Programming Guide v5.02 Patch2

[2] Zensys, INS10795, Instruction, 400 Series Z-Wave Single Chip Programming Mode

[3] Zensys, INS10579, Instruction, Programming the ZW0102 Flash and Lock Bits