TB3204

DMX-512 using the UART with Protocol Support

Introduction

Author: Ashish Makthal, Microchip Technology Inc.

DMX-512 is an industry standard protocol that is commonly used to control stage lighting and theatrical effects like fog machines, and is commonly used for color changing LEDs. The latest 8-bit PIC® microcontrollers offer a new UART module that has built-in support for DMX-512 protocol. This module also supports other protocols such as DALI and LIN.

Overview

This publication discusses the capabilities of the new UART module when operating in the DMX mode. Both the Controller and Receiver modes are supported in this module. This publication also shows how to use the Direct Memory Access (DMA) module with the UART module, which helps reduce CPU load. There are code snippets in this document that will help in setting up the module.

Table of Contents

Introduction....1

Overview....1

1. DMX-512 Protocol.... 3
   1.1. Advantages of UART with Protocol Support.... 3
2. UART with Protocol as DMX Master....5
   2.1. Setup for DMX Controller.... 5
3. UART with Protocol as DMX Receiver.... 7
   3.1. Setup for DMX Receiver....7
4. UART with Protocol + DMA.... 9
   4.1. DMX Controller with DMA.... 10
   4.2. DMX Receiver with DMA.... 11
5. Conclusion....12

The Microchip Web Site.... 13

Customer Change Notification Service....13

Customer Support.... 13

Microchip Devices Code Protection Feature.... 13

Legal Notice....14

Trademarks.... 14

Quality Management System Certified by DNV....15

Worldwide Sales and Service....16

1. DMX-512 Protocol

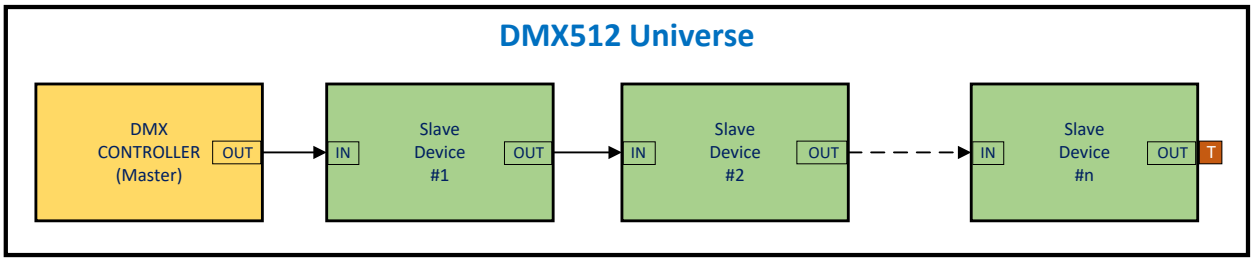
A DMX-512 system consists of a universe which is comprised of 512 channels. Each channel can have any value between 0-255. A DMX packet mainly consists of a burst of data (up to 512 data bytes). The position of the byte in the data stream determines which device in the DMX universe will respond to it. This protocol has a fixed baud rate of 250,000 bits per second. The data format for the protocol is defined as one start bit, eight data bits (LSb first), two stop bits and no parity bit.

As a DMX controller, the device is supposed to send a start code before the 512 data bytes. These framed bytes are commonly known as slots. The first frame, which consists of the start code, is called slot 0 and the next frame is slot 1 and so on until slot 512.

Tip: The start code is usually '0x00' for normal operation. Alternate start codes can be used to perform other enhanced functionality. Refer to the DMX specification for more detail about alternate start code implementations.

In a DMX universe, all the slave devices are connected in a daisy chain. Each slave device has a "DMX IN" connector and a "DMX OUT" connector. The controller (master) device usually has just a "DMX OUT" connector. A terminator is specified in the DMX specification that is to be connected to the final "DMX OUT". Refer to the figure below for a basic block diagram of a DMX universe.

Figure 1-1. Block Diagram of a DMX Universe



DMX protocol data is transmitted over a differential pair of lines using the RS-485 voltage levels. The DMX specification mentions that XLR-5 type connectors are to be used for the system connections. Refer to the DMX specification for more detail about cabling and other electrical parameters.

Tip: The specification documents for the DMX protocol and the cabling requirements can be found on the ESTA Technical Standards Program website.

1.1 Advantages of UART with Protocol Support

The new UART with protocol support module is helpful in many ways for handling the flow of data for the specific protocols. Figure 1-2 is a timing diagram for a DMX message. The individual events have specific timing requirements as defined in the DMX specifications. The UART module with protocol support will automatically produce the correct sequence of events with the timing requirements. The user software does not have to account for maintaining the timings for these events.

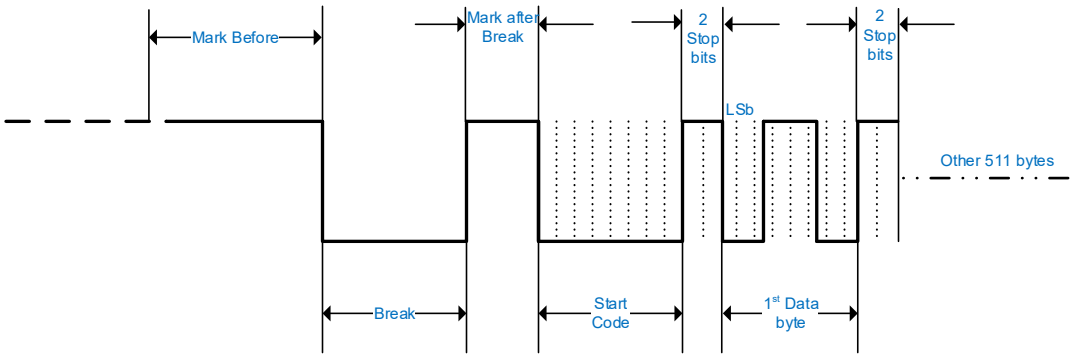
When used as a DMX receiver, the module can detect a break condition and determine if it is valid, and then start receiving the DMX packets. It will ignore the DMX packet if the break event timings are not met.

Also, as a DMX receiver only a few bytes in the stream of 512 data bytes might be significant. The UART module can be configured to receive a subset of the full DMX message and will only produce receive interrupts when the necessary bytes are received. This eliminates the need for the software to handle unnecessary interrupts and keep track of the number of bytes received.

Several of the newer 8-bit PIC microcontrollers feature the DMA module, which can be used in conjunction with the UART to transmit or receive data independently of the CPU. This frees up the CPU to handle other important tasks while the DMA handles the DMX data packet.

Figure 1-2. Timing Diagram

Description	Minimum (µs)	Maximum (µs)	Typical (µs)
Break	92	-	176
Mark after Break	12	<1,000,000	-
Bit Time	3.92	4.02	4
DMX512 packet	1204	1,000,000	-



2. UART with Protocol as DMX Master

As mentioned in the previous section, the UART with protocol helps simplify user firmware by managing the timing requirements for the DMX protocol. User firmware needs to configure the module in DMX mode, setup the baud rate generator for 250k baud based on the system clock settings, and program the number of bytes to be sent in the DMX packet.

The start of a new DMX packet is signified by the Break condition, followed by a logical one referred to as the Mark After Break (MAB) condition. The Break condition serves as a framing reference for the DMX receivers that may have lost synchronization. The UART module will produce the break, followed by the MAB, as soon as the first byte is loaded in the Transmit Buffer (UxTXB). The number of data bytes to be sent after the start code is defined by the value in the UART Parameter 1 (UxP1) register. User firmware should load this register at a value of one less than the total number of data bytes to be sent excluding the start code. Refer to the code examples below for more details.

After the required number of bytes have been transmitted, the UART module will produce two stop bits on the bus, which signifies the end of a DMX packet. The line is then held at the Mark (logical one) condition until the next Break condition is produced. This state of logical one is called the Mark Before.

To start the next DMX packet, the user firmware has to load the UxTXB register. There is no need to reconfigure the module unless the next packet is of a different size.

2.1 Setup for DMX Controller

The UART with protocol module can be configured as a DMX controller with the following settings:

- MODE<3:0> = `1010` - Selects the DMX mode
- TXEN = `1` - Enables the transmitter
- RXEN = `0` - Disables the receiver
- TXPOL = `0` - Selects regular polarity
- STP<1:0> = `10` - for 2 stop bits
- UxP1 = One less than the number of bytes to transmit (excluding the start code)
- UxBRGH:L = Value to achieve 250K baud rate
- RxyPPS = TX pin output code
- ON = `1`

Tip: Complete working demo for the DMX controller can be found on the MPLAB® Xpress Cloud-Based IDE platform. See the MPLAB Xpress Code Examples.

Example 2-1. Code snippet for setup as DMX Controller

Note: This code example is written for PIC18F25K42. The PPS setting might vary for different devices.

```
void UART_Initialize(){
    //UART module settings
    U1CON0bits.MODE = 0b1010;   //Select DMX mode
    U1CON0bits.TXEN = 1;        //Enable transmitter
    U1CON0bits.RXEN = 0;        //Disable receiver
    U1CON2bits.TXPOL = 0;       //Standard polarity, TX pin will idle high
    U1CON2bits.STP = 0b10;      //2 stop bits

    //DMX data packet settings
    U1P1 = MAXCHANNELS-1;       //Total number of data bytes - 1
    U1P2 = 0x00;                //Not used in DMX controller
    U1P3 = 0x00;                //Not used in DMX controller

    // Baud rate generator settings
    U1CON0bits.U1BRGS = 1;      //High speed baud generation
    U1BRG = 0x3F;               //Value for U1BRG for Fosc = 64MHz

    //PPS settings for TX functionality on pin RC6
    ANSELCbits.ANSELC6 = 0;     //Make RC6 a digital I/O
    TRISCbits.TRISC6 = 0;       //Make RC6 an output
    RC6PPS = 0b010011;          //Assign TX functionality to RC6

    U1ON = 0x01;                //Turn on UART module
}
```

3. UART with Protocol as DMX Receiver

The UART with protocol can be configured as a DMX receiver and the module can detect and respond to only valid break + Mark After Break (MAB) conditions. This helps the microcontroller to idle or attend to other high priority interrupts while it waits for a break condition. A break received interrupt is available, which will trigger only if the correct timing requirements are met for the break condition.

The first byte following a break + MAB condition is always received in the UART Receive Buffer (UxRXB). This is the slot 0 data or commonly known as the start code of the DMX packet. User firmware can read this byte out of the UxRXB register and branch away from normal reception, if it is a special start code.

Based on the application, the receiver would be interested in only a small range of data in the complete DMX packet. The UART with protocol helps simplify user firmware by having a configurable address range where the module will ignore all the data bytes outside that range. The address range is defined by the UART Parameter 2 (UxP2: start address) and UART Parameter 3 (UxP3: end address) registers. The module will produce receive interrupts only for the range defined by these registers. This helps reduce unnecessary software overhead to handle unwanted interrupts.

The module can be configured to wait for two stop bits at the end of the packet and verify that two stop bits were received by setting the STP<1:0> bits of UxCON2 register to '0b10'.

User firmware does not have to update the configuration to receive the next DMX packet, unless the address range is changing.

3.1 Setup for DMX Receiver

The UART with protocol module can be configured as a DMX receiver with the following settings:

- MODE<3:0> = 1010 - Selects the DMX mode
- TXEN = 0 - Disables the transmitter
- RXEN = 1 - Enables the receiver
- RXPOL = 0 - Selects regular polarity
- STP<1:0> = 10 for verifying the reception of 2 stop bits
- UxP2 = Address of first byte to receive
- UxP3 = Address of last byte to receive
- UxBRGH:L = Value to achieve 250K baud rate
- UxRXPPS = Code for desired input pin
- Selected input pin should be made a digital input by clearing the corresponding ANSEL bit
- ON = 1

Tip: Complete working demo for the DMX receiver can be found on the MPLAB® Xpress Cloud-Based IDE platform. See the MPLAB Press Code Examples.

Example 3-1. Code snippet for setup as DMX receiver

Note: This code example is written for PIC18F25K42. The PPS setting might vary for different devices.

```
void UART_Initialize(){
    //UART module settings
    U1CON0bits.MODE = 0b1010;   //Select DMX mode
    U1CON0bits.TXEN = 0;        //Disable transmitter
    U1CON0bits.RXEN = 1;        //Enable receiver
    U1CON2bits.RXPOL = 0;       //Standard polarity, RX pin will idle high
    U1CON2bits.STP = 0b10;      //Recevie and verify 2 stop bits

    //DMX data packet settings
    U1P1 = 0x00;                //Not used in DMX receiver
    U1P2 = 0x10;                //Address of first byte of interest
    U1P3 = 0x20;                //Address of last byte of interest

    // Baud rate generator settings
    U1CON0bits.U1BRGS = 1;      //High speed baud generation
    U1BRG = 0x3F;               //Value for U1BRG for Fosc = 64MHz

    //PPS settings for RX functionality on pin RC7
    ANSELCbits.ANSELC7 = 0;     //Make RC7 a digital I/O
    TRISCbits.TRISC7 = 0;       //Make RC7 an input
    U1RXPPS = 0b010111;         //Assign RX functionality to RC7

    U1ON = 0x01;                //Turn on UART module
}
```

4. UART with Protocol + DMA

Several of the newer 8-bit PIC microcontrollers feature the Direct Memory Access (DMA) module, which can help move blocks of data without CPU involvement. The DMX-512 protocol is well suited for DMA based operation as there can be up to 513 bytes (including the start code) that need to be transmitted. Generally, these bytes are stored in a buffer with the microcontroller and using the DMA module, these bytes can be moved into the UxTXB without any involvement of the CPU. The DMA can be used at the receiver end, and to transfer the data bytes defined by the address range into a buffer for further processing. This makes the application completely core independent. The CPU is required only during the initial setup of the module.

Note: Refer to TB3164 for more detail about the DMA module.

4.1 DMX Controller with DMA

In the case of a DMX controller, the microcontroller will send up to 513 bytes (including the start code) from a preloaded buffer. The start code should be included as the first byte of the buffer. This makes the complete DMX packet transmission automated through the DMA. The DMA module is setup to move one byte of data at a time from the buffer to the UxTXB register. The DMA operation can be triggered by the UART Transmit Interrupt Flag (UxTXIF). Every time a byte is transmitted out of the buffer, the DMA is triggered to move the next data byte. The DMA module can be configured to stop or restart at the end of the DMX packet. The source address will increment in this case, but the destination address remains static.

The example code below is for the DMA setup to transfer 300 bytes of data into the U1TXB register. Note that the UART configuration is still required as mentioned in the section 2.1 Setup for DMX Controller. The firmware required to load a byte of data into the U1TXB register is eliminated using the DMA module.

Example 4-1. Code snippet for DMA setup to work with DMX Controller

Note: This code example is written for PIC18F25K42. Minor edits might be needed to use it for other devices that feature the DMA module.

```
void DMA1_Initialize(void)
 {
     DMA1SSA = &TX_DATA;       //set source start address
     DMA1DSA = &U1TXB;         //set destination start address
     DMA1CON1 = 0x03;          //DMODE=00|DSTP=0|SMR=00|SMODE=01|SSTP=1
     DMA1SSZ = 0x012C;         //set source size = 300 bytes
     DMA1DSZ = 0x0001;         //set destination size = 1 byte
     DMA1SIRQ = 0x1C;          //set DMA Transfer Trigger Source = U1TX
     DMA1AIRQ = 0x00;          //set DMA Transfer abort Source

     PIR2bits.DMA1DCNTIF = 0;  //clear Destination Count Interrupt Flag
     PIR2bits.DMA1SCNTIF = 0;  //clear Source Count Interrupt Flag
     PIR2bits.DMA1AIF = 0;     //clear abort Interrupt Flag
     PIR2bits.DMA1ORIF = 0;    //clear overrun Interrupt Flag
     PIE2bits.DMA1DCNTIE = 0;  //disable Destination Count Interrupt
     PIE2bits.DMA1SCNTIE = 0;  //disable Source Count Interrupt
     PIE2bits.DMA1AIE = 0;     //disable abort Interrupt
     PIE2bits.DMA1ORIE = 0;    //disable overrun Interrupt

     asm("BCF INTCON0,7");     //disable Global Interrupts
     asm ("BANKSEL PRLOCK");   //
     asm ("MOVLW 0x55");       //
     asm ("MOVWF PRLOCK");     //Arbiter Priority lock
     asm ("MOVLW 0xAA");       //sequence
     asm ("MOVWF PRLOCK");     //
     asm ("BSF PRLOCK, 0");    //
     asm("BSF INTCON0,7");     //enable Global Interrupts
     DMA1CON0 = 0x80;          //EN=1|SIRQEN=0|DGO=0|xx|AIRQEN=0|x|XIP=0
 }
```

Note: User firmware can set the SIRQEN bit whenever the DMA transaction needs to start. This can be done by the following line of code.

```
DMA1CON0bits.SIRQEN = 1;
```

4.2 DMX Receiver with DMA

In the case of DMX receiver, the microcontroller will need to receive a range of data from the complete DMX packet. As mentioned earlier, the UART module is capable of filtering unwanted data bytes, but the firmware still needs to read the necessary bytes from the UxRXB register. Using the DMA module, the received data bytes can be transferred to a buffer in the RAM area without the involvement of the CPU.

The DMA module is setup to read the incoming byte from the UxRXB register and transfer it to a location in the RAM area. The destination address will increment in this case, but the source address remains static. The DMA operation can be triggered by the UART Receive Interrupt Flag (UxRXIF). When a byte is received into the buffer, the DMA is triggered to move it to the buffer. Note that the receive interrupt will set only for the valid address range defined in the UART configuration. The DMA module can be configured to stop or restart every time the destination counter reloads.

The example code below is for the DMA setup to receive 16 bytes of data starting from address 20. Note that the start code is always received and transferred into the RAM buffer. So the DMA count should be one more than the number of required bytes. Note that the UART configuration is still required, as mentioned in section 3.1 Setup for DMX Receiver.

Example 4-2. Code snippet for DMA setup to work with DMX Recevier

Note: This code example is written for PIC18F25K42. Minor edits might be needed to use it for other devices that feature the DMA module.

```
void DMA1_Initialize(void)
{
    DMA1SSA = &U1RXB;          //set source start address
    DMA1DSA = &RX_DATA;        //set destination start address
    DMA1CON1 = 0x60;           //DMODE=01|DSTP=1|SMR=00|SMODE=00|SSTP=0
    DMA1SSZ = 0x0001;          //set source size = 1 byte
    DMA1DSZ = 0x0011;          //set destination size = 17 bytes
    DMA1SIRQ = 0x1B;           //set DMA Transfer Trigger Source = U1RX
    DMA1AIRQ = 0x00;           //set DMA Transfer abort Source

    PIR2bits.DMA1DCNTIF = 0;  //clear Destination Count Interrupt Flag
    PIR2bits.DMA1SCNTIF = 0;  //clear Source Count Interrupt Flag
    PIR2bits.DMA1AIF = 0;     //clear abort Interrupt Flag
    PIR2bits.DMA1ORIF = 0;    //clear overrun Interrupt Flag
    PIE2bits.DMA1DCNTIE = 0;  //disable Destination Count Interrupt
    PIE2bits.DMA1SCNTIE = 0;  //disable Source Count Interrupt
    PIE2bits.DMA1AIE = 0;     //disable abort Interrupt
    PIE2bits.DMA1ORIE = 0;    //disable overrun Interrupt

    asm("BCF INTCON0,7");     //disable Global Interrupts
    asm ("BANKSEL PRLOCK");   //
    asm ("MOVLW 0x55");       //
    asm ("MOVWF PRLOCK");     //Arbiter Priority lock
    asm ("MOVLW 0xAA");       //sequence
    asm ("MOVWF PRLOCK");     //
    asm ("BSF PRLOCK, 0");    //
    asm("BSF INTCON0,7");     //enable Global Interrupts
    DMA1CON0 = 0x80;          //EN=1|SIRQEN=0|DGO=0|xx|AIRQEN=0|x|XIP=0
 }
```

Note: User firmware can set the SIRQEN bit whenever the DMA transaction needs to start. This can be done by the following line of code.

```
DMA1CON0bits.SIRQEN = 1;
```

5. Conclusion

This technical brief provides an overview of using the UART with the protocol support module in the DMX mode. It explains how the new module is beneficial in handling the timing requirements for the DMX protocol. The sample code snippets serve as a walk-through on how to get started in building an application that uses the DMX module. This technical brief also shows how the DMA module can be setup to transmit and receive data without the involvement of the CPU. For more information about this module, refer to the device data sheet.

Tip: Refer to TB3200 for details about using the UART module for DALI protocol

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks

ISBN: 978-1-5224-3378-1

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-67-3636 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-7289-7561 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820