Comparing CAN and ECAN Modules

Author: Caio Gübel
Microchip Technology Inc.

With the arrival of the PIC18FXX8X family of microcontrollers featuring the Enhanced Control Area Network (ECAN) module, designers can now choose between the original CAN module present in the PIC18FXX8 family and the new ECAN-enabled parts.

This application note presents communication time comparisons along with conclusions and recommendations intended to help designers find the best CAN-based solution given an application and available parts.

These comparisons and related recommendations are based on the following rules:

1. Code processing time measured in number of instructions (assembly language)
2. Transmission and reception times measured in bit time

CAN INTERFACE CHARACTERISTICS

- Implementation of the CAN protocols: CAN 1.2, CAN 2.0A and CAN 2.0B
- Standard and extended data frames
- 0-8 bytes data length
- Programmable bit rate up to 1 Mbit/sec
- Support for remote frames
- Double-buffered receiver with two prioritized received message storage buffers
- Six full (standard/extended identifier) acceptance filters; two associated with the high priority receive buffer and four associated with the low priority receive buffer
- Two full acceptance filter masks, one each associated with the high and low priority receive buffers
- Three transmit buffers with application specified prioritization and abort capability
- Programmable wake-up functionality with integrated low-pass filter
- Programmable Loopback mode supports self-test operation
- Signaling via interrupt capabilities for all CAN receiver and transmitter error states
- Programmable clock source
- Programmable link to timer module for time-stamping and network synchronization
- Low-power Sleep mode

ECAN INTERFACE CHARACTERISTICS

- Implementation of the CAN protocols: CAN 1.2, CAN 2.0A and CAN 2.0B
- DeviceNet™ data bytes filter support
- Standard and Extended data frames
- 0-8 bytes data length
- Programmable bit rate up to 1 Mbit/sec
- Fully backward compatible with PIC18XX8 CAN module
- Three modes of operation:
  - Mode 0 – Legacy mode
  - Mode 1 – Enhanced Legacy mode with DeviceNet support
  - Mode 2 – FIFO mode with DeviceNet support
- Support for remote frames with automated handling
- Double-buffered receiver with two prioritized received message storage buffers
- Six buffers programmable as RX and TX message buffers
- 16 full (standard/extended identifier) acceptance filters that can be linked to one of four masks
- Two full acceptance filter masks that can be assigned to any filter
- One full acceptance filter that can be used as either an acceptance filter or acceptance filter mask
- Three dedicated transmit buffers with application specified prioritization and abort capability
- Programmable wake-up functionality with integrated low-pass filter
- Programmable Loopback mode supports self-test operation
- Signaling via interrupt capabilities for all CAN receiver and transmitter error states
- Programmable clock source
- Programmable link to timer module for time-stamping and network synchronization
- Low-power Sleep mode

For more information on CAN and ECAN, refer to **Appendix A: “References”**.

TRANSMISSION LATENCY

CAN implements a bus access method called Carrier Sense Multiple Access with Collision Detection and Arbitration on Message Priority (CSMA/CD+AMP). In this method, transmission conflicts are solved based on the priority of the message, defined by its ID (the smaller the ID number, the higher the priority).

This method, although reliable, presents a variable total message transmission time that, in general, doesn't pose a problem in soft real-time systems. However, in hard real-time systems that require a deterministic latency, such behavior may cause dead times to be missed. The analysis of the worst-case latency is beyond the scope of this text (see **Appendix B: "CAN Analysis"** for references on the subject). The simplified equation describing the process is given by the following equation:

EQUATION 1:

$$L = Qj + Qt + Tt$$

where:

L = Total Latency

Qj = Queuing jitter, the maximum time spent by the sending task. It represents the time spent to load the message in the CAN peripheral and request the transmission.

Qt = Queuing time, the effective maximum time spent to queue the message. It comprises the time to complete the transmission of a message that already won a previous arbitration cycle, plus the transmission time of all messages with higher priority, plus all bus error signaling/recovery.

Tt = Effective transmission time, the time to send the frame through the bus.

From this we can devise some simple, yet very important results:

1. The time spent to load the message in the peripheral may become relevant in high-speed busses (close to 1 Mbit/sec) and therefore, the larger the number of available TX buffers, the lower the time spent managing transmission buffers. In general, it is considered that three transmission buffers are the minimum recommended.
2. The best approach for scheduling the transmission implements a deadline monotonic approach, where the shortest deadline is assigned to the highest priority and so forth, in decreasing order of priority. The Microchip application note, "*AN853, PIC18XXX8 CAN Driver with Prioritized Transmit Buffer*" (DS00853), presents an interesting implementation of a prioritized transmission scheme.
3. The message total size (including stuff bits) shall be reduced as much as possible, therefore, careful selection of the message ID and message content is pivotal.

   Appendix B: "CAN Analysis" provides a reference to an interesting approach of control of priority and data transformation is used to minimize stuffing.

RECEPTION REACTION OF THE APPLICATION PROGRAM

The CAN protocol ensures the proper distribution of a message but not the proper processing of the message by the microcontroller to which it is intended. Therefore, it is crucial to ensure that the microcontroller receiving the message is capable of processing it so that no message is overwritten.

There are three basic approaches to this situation:

1. Increase the speed (clock of the processor)
2. Decrease the bus speed
3. Increase the number of available reception buffers

The first approach implies that not only can the microcontroller be run with a higher clock, but also that the power consumption and the noise generated (EMI related) is inside the specifications for the project. In some situations, this is a possible approach, but it requires careful analysis and in some cases, recertification of the hardware.

The second approach is less probable, as in many cases the bus speed is tied to external factors beyond the control of the designer, such as maximum latency for critical messages and legacy HW/Environment.

The third approach implies the existence of more buffers, or the possibility of upgrading the microcontroller to a part with more buffers.

To evaluate the impact of having a different number of buffers present in the CAN module and the ECAN module and also to take in account the effect of the speed of the processor, some calculations were done based on the following conditions:

- Main Clock: 16, 25, 33 and 40 MHz (16 MHz is the minimum frequency necessary to achieve 1 Mbit/sec)
- Bus Speed: 1 Mbit/sec
- RTR frame time (no data): 47 and 67 µs
- Response message transmit time: 130 and 154 µs

Table 1 relates the frequency of the microcontroller and the processing time in microseconds for reception and transmission in both standard and extended frames for CAN and ECAN modules. (Arbitrary 25 instruction delay was added to account for branch and interrupt processing times.)

TABLE 1: CAN AND ECAN MICROCONTROLLER FREQUENCY AND PROCESSING TIMES

Freq (MHz)	CAN				ECAN			
	RX		TX		RX		TX	
	Standard Frame	Extended Frame	Standard Frame	Extended Frame	Standard Frame	Extended Frame	Standard Frame	Extended Frame
16	43.0	47.3	56.8	59.5	58.1	67.2	76.8	85.8
25	27.5	30.2	36.3	38.1	37.2	43	49.1	54.9
33	20.8	22.9	27.5	28.8	28.2	32.6	37.2	41.6
40	17.2	18.9	22.7	23.8	23.2	26.9	30.7	34.3

Table 2 shows the processing and consumption time for both a CAN-enabled part and an ECAN-enabled part when a RTR message is received and responds back an 8-byte message (both Standard frame).

TABLE 2: STANDARD FRAME RTR MESSAGE PROCESSING AND CONSUMPTION TIMES

Freq (MHz)	CAN	ECAN	CAN	ECAN	CAN	ECAN	CAN	ECAN	CAN	ECAN	CAN	ECAN	CAN	ECAN
	RX0 (µs)		**PRX0** (µs)		**TX0** (µs)		**PTX0** (µs)		**Total Time** (µs)		P_{Duty}		B_{Duty}	
16	47	47	43	0	130	130	57	0	278	178	36%	0%	64%	99%
25	47	47	28	0	130	130	36	0	242	178	26%	0%	73%	99%
33	47	47	21	0	130	130	28	0	227	178	22%	0%	78%	99%
40	47	47	17	0	130	130	27	0	222	178	20%	0%	80%	99%

Legend: RX0 is the bus time of the RTR message.
PRX0 is the processing time of the received message.
PTX0 is the processing time of the message to be sent back.
TX0 is the bus time of the message sent back.
P_{Duty} is the processing time as a percent of the total time.
B_{Duty} is the bus time as a percent of the total time.

Note 1: 1 µs minimum extra time given between the reception and the transmission.

Table 3 shows the processing and consumption time for both a CAN-enabled part and an ECAN-enabled part when a RTR message is received and responds back an 8-byte message (both Extended frame).

TABLE 3: EXTENDED FRAME RTR MESSAGE PROCESSING AND CONSUMPTION TIMES

Freq (MHz)	CAN	ECAN	CAN	ECAN	CAN	ECAN	CAN	ECAN	CAN	ECAN	CAN	ECAN	CAN	ECAN
	RX0 (μs)		**PRX0** (μs)		**TX0** (μs)		**PTX0** (μs)		**Total Time** (μs)		P_{Duty}		B_{Duty}	
16	67	67	47	0	154	154	60	0	329	222	33%	0%	67%	100%
25	67	67	30	0	154	154	38	0	290	222	23%	0%	76%	100%
33	67	67	23	0	154	154	29	0	274	222	19%	0%	81%	100%
40	67	67	19	0	154	154	19	0	265	222	16%	0%	83%	100%

Legend: RX0 is the bus time of the RTR message.
PRX0 is the processing time of the received message.
PTX0 is the processing time of the message to be sent back.
TX0 is the bus time of the message sent back.
P_{Duty} is the processing time as a percent of the total time.
B_{Duty} is the bus time as a percent of the total time.

Note 1: 1 μs minimum extra time given between the reception and the transmission.

The data analysis shows that the CAN module response process to a RTR message can consume from 16%, up to 36% of the overall processing time against 0% of the ECAN part and may take up to 56% more time, which would reduce the effective data baud rate to 640 Kbits/sec in a 1 Mbit/sec based bus.

A similar analysis can be made for the case of systems processing several back-to-back RTR messages. An important point is that during the additional time required by the application to process the RTR message, another message with a lower priority can gain access to the bus and win the arbitration, causing an inversion of priority which would delay the response to an RTR even further. An example of such an event would result in the following pattern: 1 Mbit/sec, micro-controller at 16 MHz, Standard frame, RTR and 8-byte messages.

As shown in Table 4, in this case the effective bus data rate fell to 430 Kbits/sec in a 1 Mbit/sec bus system.

TABLE 4: RTR MESSAGE PROCESSING WITH PRIORITY INVERSION

Frequency (MHz)	RX0	PRX0	PTX0	TX1	TX0	Time	P_{Duty}	B_{Duty}
16	47	43	57	130	130	408	25%	43%

Legend: RX0 is the bus time of the RTR message.
PRX0 is the processing time of the received message.
PTX0 is the processing time of the message to be sent back.
TX1 is the message with lower priority that gained access to the bus.
TX0 is the bus time of the message sent back.
P_{Duty} is the processing time as a percent of the total time.
B_{Duty} is the bus time as a percent of the total time.

Note 1: 1 μs minimum extra time given between the reception and the transmission.

Table 5 illustrates using the situation described in the previous example (standard ID size, 11 bits):

TABLE 5: RECEPTION OF LARGE AMOUNTS OF DATA

Frequency (MHz)	RX0 (µs)	PRX0 (µs)	P_{Duty}	B_{Duty}
16	130	43	36%	64%

Legend: RX0 is the bus time of the RTR message.
PRX0 is the processing time of the received message.
P_{Duty} is the processing time as a percent of the total time.
B_{Duty} is the bus time as a percent of the total time.

Note 1: 1 µs minimum extra time given between the reception and the transmission.

We can therefore calculate:

TABLE 6: RECEPTION OF DATA SUBJECTED TO LOWER PRIORITY MESSAGES GAINING BUS ACCESS

Frequency (MHz)	RX0	PRX0	nTX1(1)	TX1(2)	Time	P_{Duty}	B_{Duty}
16	130	43	0	0	1385	25%	75%
16	130	43	1	130	1515	23%	69%
16	130	43	2	260	1645	21%	63%
16	130	43	3	390	1775	19%	59%

Legend: RX0 is the bus time of the RTR message.
PRX0 is the processing time of the received message.
nTX1 is the number of lower priority messages that gained access to the bus.
TX1 is the total lower priority message transmission time.
P_{Duty} is the processing time as a percent of the total time.
B_{Duty} is the bus time as a percent of the total time.

Note 1: 1 µs minimum extra time given between the reception and the transmission.

The total reception times in this situation go from 1385 µs, up to 1775 µs, with considerable processing required against a constant reception time of about 1050 µs for the ECAN module set with 8 reception buffers and no processing.

A second situation, where the number of available reception buffers may cause impact, occurs whenever large quantities of data are transferred. A typical example of such a situation occurs when a CAN-based bootloader is developed to update the firmware of a microcontroller. In this case, typically, the program memory is programmed in 64-byte portions.

A third relevant situation occurs whenever a system needs to filter a relatively high number of messages from the total number of messages sent into the bus.

A CAN-based system can filter up to 6 messages against a total of 16 messages in ECAN devices. Therefore, if a CAN-based system wants to filter more than 6 messages, a firmware filter must be implemented.

Assuming a "switch-like" implementation, the processing time of a message can be modeled as a variable one, as shown in Equation 2:

EQUATION 2:

$$(30 + 20) \le T_m \le (20 + 30((N_i - 6) + 1)/2)$$

where:

T_m = Total processing time in number of instructions,

N_i = number of messages of interest,

20 accounts for the function calls and interruption handling routines and

30 accounts for the switch processing time.

To show the impact caused by a firmware-based filtering in the processing time, a small analysis was made. The main assumptions were:

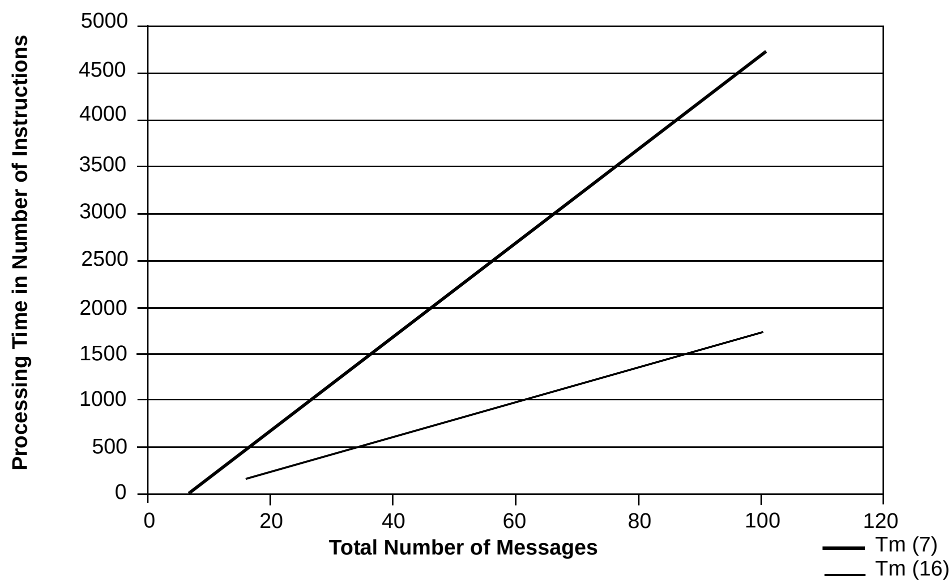
- The total number of different messages present in the bus ranges from 7, up to 100
- The probability of these messages is flat (i.e., all messages have the same chance of being transmitted)
- The system is interested in a variable number of messages, from 7 (one more than a CAN-enabled device can handle in hardware) up to 16 messages (10 more messages than the device can handle in hardware)
- The filter processing time was taken as the average from Equation 2 for a given number of messages of interest (7 to 16)

Figure 1 presents the results for two extreme points:

1. When the microcontroller is looking for seven messages of interest in a universe of seven to one hundred total equally possible messages
2. When the microcontroller is looking for sixteen messages of interest in a universe of sixteen to one hundred total equally possible messages

Figure 1 also shows that an increase in the number of total messages in a bus can dramatically increase the total amount of time spent by the microcontroller to process in firmware the extra messages, even more when compared with an ECAN-enabled device that can handle in hardware all 16 messages without any code overhead.

FIGURE 1: AVERAGE Tm MESSAGE PROCESSING TIME



CONCLUSIONS AND RECOMMENDATIONS

Both CAN and ECAN modules introduced by Microchip present stable and reliable platforms for the development of CAN-based applications; however, each one of these platforms is better suited for a different environment. The following table summarizes the main characteristics of both solutions:

TABLE 7: MAIN CHARACTERISTICS OF CAN AND ECAN

	Transmission – Low Traffic (duty < 10%)	**Transmission – High Traffic (duty > 10%)**
CAN	Good – Excellent Trade-Off Processing Time X Code Size	Higher processing times and possible delays due to priority inversions
ECAN	Almost no processing time but larger code(1)	Good – Up to 9 transmission buffers in Modes 1 and 2 and low probability of delays due to priority inversions
	Reception – Low Traffic (duty < 10%)	**Reception – High Traffic (duty > 10%) including RTR Messages**
CAN	Good – As in transmission, an excellent Trade-Off Processing Time X Code Size	High processing times and possible overload due to low number of buffers (2) and firmware-based filtering
ECAN	Almost no processing time but larger code(1)	Good – Up to 8 reception buffers in Modes 1 and 2. Low overhead in normal message reception and automatic RTR treatment (non-firmware based).

Note 1: Implementation dependent. The processing time is a function of the code size and the adopted techniques.

From Table 7 we can derive the following conclusions.

CAN-enabled devices are suitable for:

- Low traffic environments
- Environments where few messages (up to 6) must be received and processed by the system
- Environments where no or few back-to-back transmissions of large quantities of information are required

ECAN, on the other hand, is recommended for:

- High traffic environments
- Environments where the system must select, receive and process a larger number of messages (typically up to 16 messages). For this type of environment, the Mode 1 (Enhanced Legacy) is very well suited as up to 16 programmable filters are available. DeviceNet™ and CanOpen are good examples of networks that can take good advantage of an ECAN module in Mode 1 (Enhanced Legacy) due to the large quantity of messages that can be handled with minimal processing overhead. Another advantage to DeviceNet networks in this mode is the use of data byte filters that further extend the filtering capabilities of the device.
- Environments where back-to-back transmissions of large quantities of information are required. In this case, the Mode 2 (FIFO mode) is particularly suitable as it handles bursts of data in a very straightforward manner. Bootloaders are a good example of such systems that may take advantage of an ECAN module in Mode 2 (FIFO), as the microcontroller can handle a complete 64 bytes simultaneous programming sequence per burst.

APPENDIX A: REFERENCES

Microchip is fully committed to supporting the designer in the development and delivery of CAN-based applications. The following reference within Microchip's web site is the focal point from which to start a search on the subject. From there, an array of application notes, reference designs and reference codes can be found.

http://www.microchip.com/1010/suppdoc/design/netdez/can/index.htm

CAN Specification: Originally developed by R. Bosch, this is the authoritative source of information on the subject, available along with a large quantity of related information from:

http://www.can.bosch.com/index.html

APPENDIX B: CAN ANALYSIS

Book

"*CAN System Engineering – From Theory to Practical Applications*". Wolfhard Lawrenz. Springer-Verlag, New York, 1997. ISBN 0387949399

Articles

"*Minimizing CAN Response Time Jitter by Message Manipulation*". Thomas Nolte, Hans Hansson and Christer Norström. Mälardalen Real-Time Research Centre, Department of Computer Engineering, Mälardalen University, Västerås, Sweden.
http://www.mrtc.mdh.se

"*Probabilistic Worst-Case Response Time Analysis for the Controller Area Network*". Thomas Nolte, Hans Hansson and Christer Norström. Mälardalen Real-Time Research Centre, Department of Computer Engineering, Mälardalen University, Västerås, Sweden.
http://www.mrtc.mdh.se

"*Using Bit-stuffing Distribution in CAN Analysis*". Thomas Nolte, Hans Hansson, Christer Norström and Sasikumar Punnekkat. Mälardalen Real-Time Research Centre, Department of Computer Engineering, Mälardalen University, Västerås, Sweden.
http://www.mrtc.mdh.se

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.

Trademarks

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company’s quality system processes and procedures are for its PICmicro® 8-bit MCUs, 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

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

San Jose
1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde
Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-22290061 Fax: 91-80-22290062

Japan
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands
P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/17/04