



AN1221 APPLICATION NOTE

A Simple Method to Drive a Triac With ST52x420 Directly From the Main

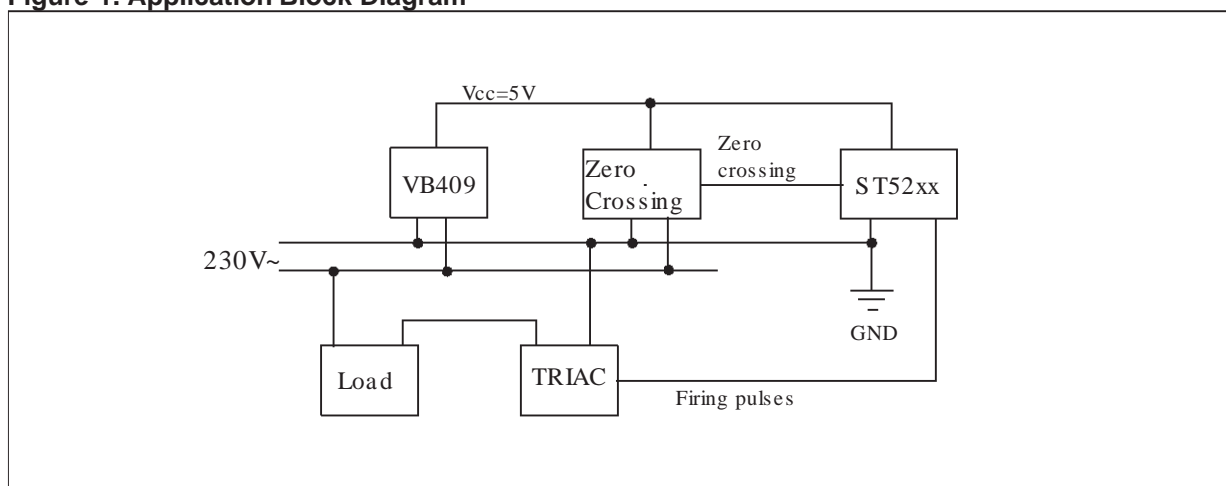
Authors: A. Cucuccio, M. Lo Presti

1. INTRODUCTION

The aim of this application note is to provide the user with a simple method to obtain a cheap circuitry for the driving of a Triac directly from the mains by using the ST52x420 microcontroller and the VB409 VIPower device acting as a 5.2V power supply.

In particular, we will describe a possible solution to drive a load in Phase Angle Partialization mode. In this case we can use the following configuration:

Figure 1. Application Block Diagram

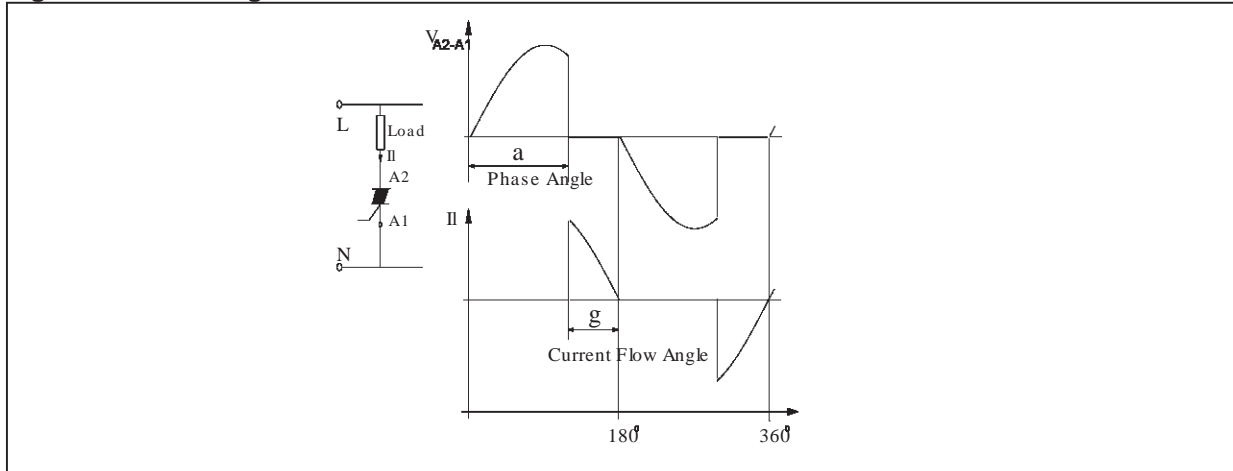


Here, we have the following blocks:

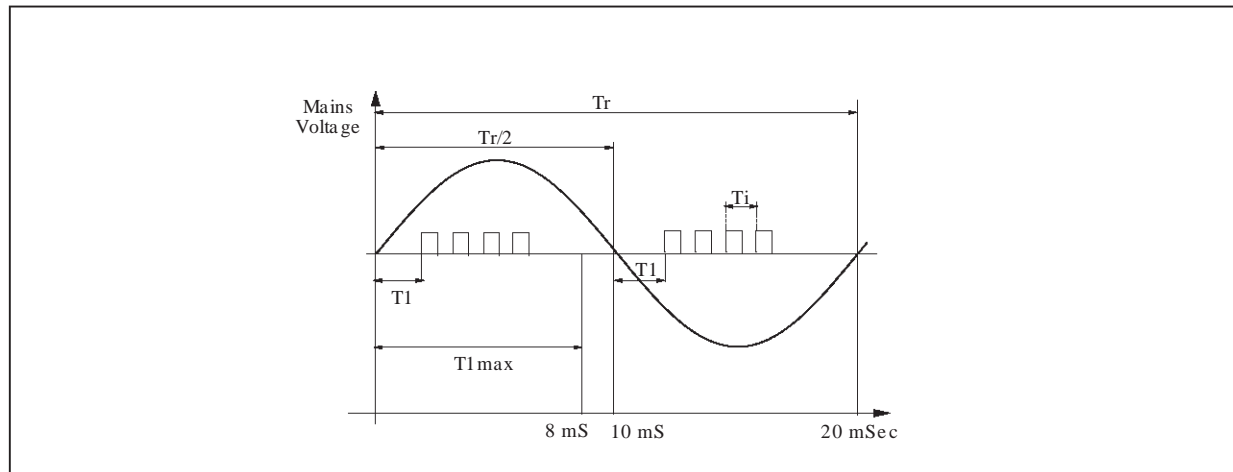
- **ST52x420:** a member of ST52 microcontroller family
- **VB409:** a VIPower device to obtain a 5.2V power supply directly from the mains
- **Zero Crossing:** a simple circuit to detect the zero crossing of the mains voltage
- **Triac:** a device working in the fourth quadrant to switch on the load
- **Load:** a generic resistive or inductive load.

In phase angle the Triac is controlled each period of the main voltage. The power transferred to the load is proportional to the Current Flow Angle γ . This kind of triac control is suitable to drive the triac with inductive load (i.e. Universal or monophasic motors). In Fig. 2 is shown the relation between the Phase Angle α and the current flow angle γ . The target is to control the Phase Angle or equivalently the time T1 (see Fig. 3).

When a motor is controlled by using this method the speed is changed by increasing or decreasing the phase angle. The Phase Angle must be synchronized with the main and it must be kept constant during the two main semi-periods. This means that, when the main voltage frequency is 50 Hz the phase angle can be changed each 20 mSec.

Figure 2. Phase Angle Partialization Mode


If the main voltage frequency is equal to 50 Hz, then T_r (see following Figure) is equal to 20 mSec. In order to avoid problems for the triac firing, when the load is inductive, several pulses must be generated. In our example 4 different pulses are used. This characteristic imposes to limit the Phase Angle or the time T_1 . Infact when T_1 is close to $T_r/2$ the triac can be fired in the second half period of the main voltage. In our example the time T_{1max} is fixed to 8.7 mSec. considering a 50Hz main voltage frequency.

Figure 3. Phase Angle Partialization Mode


To have a complete idea of the connections between the various devices we can refer to the following schematic:

In the previous schematic we have used a micropower voltage supervisor TS831 that acts as a power-on reset generator for the microcontroller and as a power failure detector.

The zero crossing circuit is realized with only two resistors and two diodes: R9 to limit the current, R3 to pull-up the zero crossing signal, D3 to cut a half-wave and D2 (a signal diode) to clamp to ground the zero crossing signal when it goes under the ground.

The VB409 is a fully protected positive voltage regulator. The device is well suited for applications powered from the AC mains and requiring a 5.2V DC regulated output voltage without galvanic isolation.

3. DEVICES CONFIGURATION

Following you will find detailed information on the devices configuration used in this application.

ST52x420 Microcontroller

The fuzzy microcontroller is used simply to drive the triac in phase angle partialization mode. The ST52x420 reads the Phase Angle to be used, by the internal A/D converter and an external potentiometer. The synchronization with the main is obtained by using the ST52x420 external Interrupt (PC0/INT). The time T1 is fixed by the timer PWM/Timer 1. The 4 pulses are generated by a software routine and using two I/O lines to obtain the gate current needed by the triac during the switch on phase.

The following steps are implemented by the ST52x420 (for more detailed information see the attached assembler code):

Step 1:

ST52x420 configuration; the Timer 1 is configured to have an output with a square waveform. The period of this signal is fixed by the prescaler value and by the Timer 1 counter value. The duty cycle is 50%. The interrupts are programmed to be generated on the falling edge of the Timer 1 output.

Step 2:

On the main program, wait for the external interrupt generated by the main voltage zero crossing.

Step 3:

When the external interrupt occurs (voltage zero crossing), inside the external interrupt routine, the desired phase angle is read by the potentiometer P1 and the A/D Channel 1. This value is stored on the RAM register 1.

The desired value is saturated, in our example it was fixed to 170. A flag is set to '0'.

The RAM register 1 is used to load the Timer 1 counter. Then the Timer 1 starts to count. The micro waits for the first PWM/Timer1 Interrupt. T1 can be calculated by using the following formula:

$$T_1 = \frac{(\text{Timer 1_Counter} \times \text{Tclk} \times \text{Prescaler})}{2}$$

Tclk is the clock master period; Prescaler is the value of the 16 bit prescaler of the PWM/Timer1. In our example Tclk=200nSec and Prescaler=512. Timer1_Counter is the value read by the potentiometer and stored on the RAM register 1. The maximum value of T1 is T1max=8.7mSec.

Step 4:

When the Timer 1 Interrupt occurs, inside the Timer 1 interrupt routines, the 4 pulses for the triac firing are generated. Considering that T1max is 8.7mSec, the maximum length of the 4 pulses must be less than 1.3 mSec.

If the flag is '0' the Timer 1 is reloaded to count 10 mSec. a flag is set to '1'. The system waits for the next Timer 1 Interrupt.

Step 5:

When the Timer 1 Interrupt occurs, the 4 firing pulses are generated. Then the system waits for the External interrupt. The algorithm restarts from the point 2.

All the previous steps are implemented in assembler. The code with a detailed description is shown in the appendix A. The Fig. 4 shows the program flow chart.

It is important to configure correctly the Timer 1 and its interrupts. For more information see the ST52x420 data sheet. The figure 5 shows the Timer1 output and the relative interrupt generation.

Figure 5. Program Flow Chart

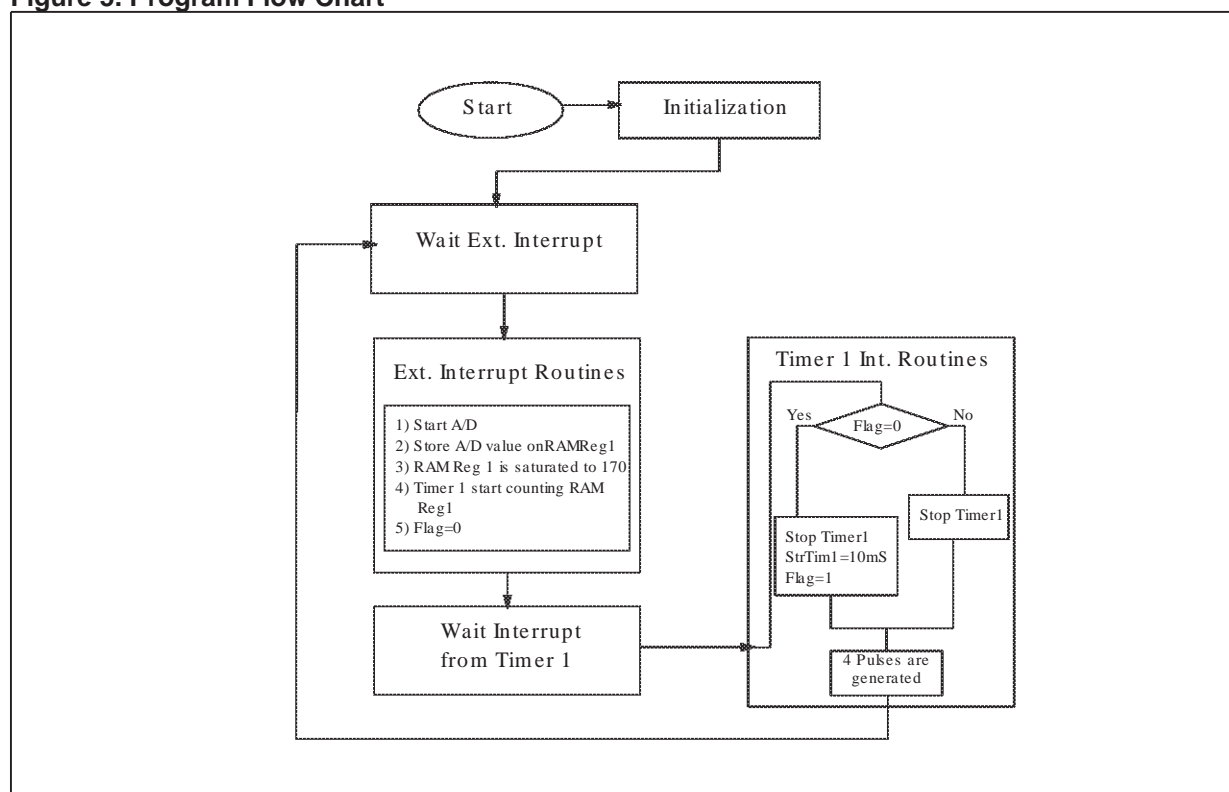
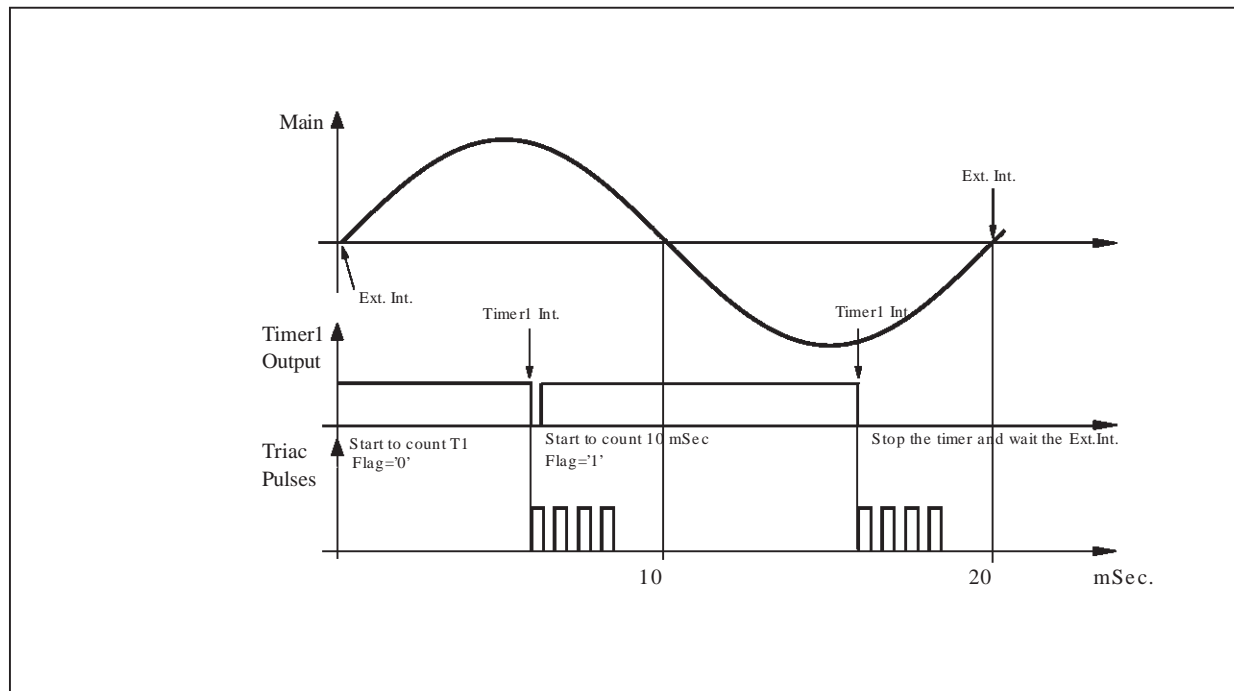


Figure 6. Waveforms and Interrupts



Power Supply by VB409

In this section we will describe how to use the VB409 to obtain a stabilized power supply for the system. The VB409 is a positive voltage regulator supplied by STMicroelectronics. VB409 provides up to 70mA output current (internally limited) at 5V. The device is well suited for applications powered from the AC mains and requiring a 5V DC regulated output voltage. The theoretical maximum current the device can source, can be derived considering that the minimum voltage on the CAP pin, at the end of the half-period in which the mains is not rectified, must be at least $V_{out}(5V) + V_{dropout}(2V)$. The maximum voltage on the CAP Pin is changed internally to 12.8 V by the VB409. During this time there is a discharge of the external capacitor at a constant current, so the voltage variation on it is linear. Thus, the theoretical maximum current the device can source is:

$$I_{max} = C \cdot \Delta V_{capmax} / (T/2)$$

In this case $T/2 = 10\text{ms}$, $\Delta V_{capmax} = 12.8 - 7 = 5.8\text{V}$, $C = 47\mu\text{F}$ and then $I_{max} = 27\text{mA}$.

Considering the current required to drive the linear part of the device, it is possible to say that $I_{max} = 23\text{mA}$. With this configuration we have measured the maximum temperature on the device (without heatsink) to be lower than 65 °C.

If we consider an external electrolytic of 100μF the maximum theoretical current is about $I_{max} = 55\text{mA}$, but in practice only 40mA must be considered because at this current value the thermal shutdown acts.

A software tool to help you with the thermal dimensioning of the VB409 by properly choosing the external capacitor and resistive divider values is available. It can be requested by electronic mail at the following address: VIPer.support@st.com. For further information on VB409, please refer to its relative data sheet.

Triac

With the power supply configuration used in our application, it is necessary to use a standard Triac in order to fire it in the fourth quadrant by applying a 5V positive pulse with respect to the ground (and therefore with respect to A1 Triac pin).

4. CONCLUSION

In this application note we propose an immediate solution to use a member of the ST52 family of microcontrollers without adding any additional component to realize the circuitry for the power supply or the Triac Driver. We suggest to use a simple zero crossing circuit if a partialization angle phase on the load is required.

APPENDIX

```
*****
;
; PROGRAM:    un_mot.asm
; PURPOSE:    Universal motor drive
;
; DATE:       12/1/2000
;
*****

*****
; This program drives an universal motor
;
*****

***** interrupt vector *****

    irq 2    TIMER_1

    irq 4    EXT_interrupt

;***** Start of program and testing of configurations *****

Start:

    ldrc 0 0      ; interrupt mask configuration
    ldcr 0 0      ; all disabled (default)

    ldrc 0 228    ; interrupt priority configuration
    ldcr 1 0      ; from the top: ADC, PWM0, PWM1, PWM2

    ldrc 0 1      ; watchdog configuration
    ldcr 2 0      ; 62.6ms count

    ldrc 0 58     ; ADC configuration
    ldcr 3 0      ; Group Chan 0,1, clock divided, power down,
                  ; continuous, stop conversion

    ldrc 0 255    ; Port A configuration
    ldcr 4 0      ; all pins in input
```


ldrc 0 0	; PWM-Timer 0 configuration
ldcr 5 0	; default settings
ldrc 0 0	; PWM-Timer 0 configuration
ldcr 6 0	; default settings
ldrc 0 0	; PWM-Timer 0 configuration
ldcr 7 0	; default settings
ldrc 0 74	; PWM-Timer 1 configuration
ldcr 8 0	; Timer mode, interrupt on falling TIMEOUT edges, ; start and reset on edge, resetted, stopped
ldrc 0 41	; PWM-Timer 1 configuration
ldcr 9 0	; square waveform, fclock/512
ldrc 0 74	; PWM-Timer 2 configuration
ldcr 10 0	; Timer mode, interrupt on falling TIMEOUT edges, ; start and reset on edge, resetted, stopped
ldrc 0 41	; PWM-Timer 2 configuration
ldcr 11 0	; square waveform, fclock/512
ldrc 0 8	; Port A mode configuration
ldcr 12 0	; port A 8 bit
ldrc 0 255	; Port B direction configuration
ldcr 13 0	; all pins input
ldrc 0 255	; Port B mode configuration
ldcr 14 0	; all pin analog
ldrc 0 1	; Port C direction configuration
ldcr 15 0	; PC0=EXT_INT, PC1-PC3=output pins
ldrc 0 7	; Port C I/O mode configuration
ldcr 16 0	; loading from RAM

;***** end configuration register *****

```
ldrc 0 0      ; Pin 1-2 port_C are low
ldpr 2 0
```

```
ldrc    0 9      ; Enable interrupt ext., timer1
ldcr    0 0
```

```
ldrc 0 62      ; ADC configuration
ldcr 3 0      ; Group Chan 0,1 clock divided, power up,
               ; continuous, stop conversion
```

```
ldrc 0 63      ; ADC configuration
ldcr 3 0      ; Group Chan 0,1 clock divided, power up,
               ; continuous, start conversion
```

```
***** Main Program *****
```

Wait:

```
waiti      ;
```

End: jp Wait

EXT_interrupt:

```
ldri 1 2      ; Read A/D converter ch. 1 and store on register1
ldrc 5 170    ; Saturation value=170
sub 5 1
jpns no_sat
ldrc 1 170
```

no_sat: jp no_tim1 ; No enable timer 1, wait next external interrupt

```
ldpr 5 1      ; Load register1 value on timer1 counter reg.
```

```
ldrc 1 69     ; PWM-Timer 1 configuration
ldcr 8 1      ; Timer mode, interrupt on falling TIMEOUT edges,
               ; start and reset on edge, setted, started
```

```
ldrc 12 1      ; reg 12 is a flag to know if is the first
                ; half wave or the second halfwave

no_tim1:
    reti

TIMER_1:

    ldrc 2 1      ; Check for the flag
    sub 2 12
    jpz f_t

    jp s_t

f_t:            ; First halfwave

    ldrc 12 0

    ldrc 2 64      ; PWM-Timer 1 configuration
    ldcr 8 2      ; Timer mode, interrupt on falling TIMEOUT edges,
                ; start and reset on edge, resetted, stopped

    ldrc 2 195     ; Load 195 on timer counter1 -> count for 10mS
    ldpr 5 2

    ldrc 2 69      ; PWM-Timer 1 configuration
    ldcr 8 2      ; Timer mode, interrupt on falling TIMEOUT edges,
                ; start and reset on edge, setted, started
    jp p_tr

s_t:            ; Second halfwave

    ldrc 2 64      ; PWM-Timer 1 configuration
    ldcr 8 2      ; Timer mode, interrupt on falling TIMEOUT edges,
                ; start and reset on edge, resetted, stopped

p_tr:
    ldrc 10 4      ; Send 4 pulses to triac
```

n_pull:

```
ldrc 2 6          ; Pin 1-2 port_C are high
ldpr 2 2
```

rit_1:

```
ldrc 11 20        ; Set the pulses lenght (high value)
```

```
dec 11
jpnz rit_1
```

```
ldrc 2 0          ; Pin 1-2 port_C are low
ldpr 2 2
```

rit_1a:

```
ldrc 11 20        ; Set the pulses lenght (low value)
```

```
dec 11
jpnz rit_1a
```

```
dec 10
jpnz n_pull
```

```
reti
```

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a trademark of STMicroelectronics
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved

FUZZYSTUDIO™ is a registered trademark of STMicroelectronics

STMicroelectronics GROUP OF COMPANIES

<http://www.st.com>

Australia - Brazil - China - Finland - France - Germany - Hong Kong India - Italy - Japan - Malaysia - Malta -
Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.