Demonstrating the Soft Detach Function With a PS/2® to USB Translator Example

Author: Reston Condit
Microchip Technology Inc.

INTRODUCTION

This technical brief describes the creation of a USB device that can enumerate as a mouse or as a keyboard, depending on which PS/2 peripheral is plugged into it. The Soft Detach provision of the PIC16C745/765 support firmware makes the creation of such a device possible. Two earlier Briefs, TB055 and TB056, describe in detail the implementation of a PS/2 to USB mouse translator and a PS/2 to USB keyboard translator, respectively. This brief will not discuss the translation of either of these devices. Rather, it focuses on the Soft Detach function and on modifying the descriptor jump table for a device with multiple sets of descriptors.

Note: This technical brief is the fifth in a series of five technical briefs. This series is meant to familiarize developers with USB. For the best understanding of USB, read the briefs in order: TB054, TB055, TB056, TB057, TB058

SOFT DETACH

The `SoftDetachUSB` command enables the PICmicro® microcontroller (MCU) to have control over when it is enumerated by the host. During the Soft Detach process, several things occur. First, the microcontroller turns off the pull-up resistor to VUSB. The firmware does this by clearing the `DEV_ATT` bit. Turning the pull-up resistor off has the effect of removing the microcontroller from the bus. After approximately 50 ms, or enough time for the host to see the device disconnect, the firmware sets `DEV_ATT` and “reconnects” the microcontroller to the bus. Soft Detach then calls the `InitUSB` command and waits for the host to re-enumerate the PICmicro MCU.

Soft Detach is one of the most useful features on Microchip's PIC16C745/765 USB microcontroller. The reason the Soft Detach command is so useful is that, in applications such as this one, no special driver needs to be created for the host. The more typical approach to this application would be to create two configurations: one for a mouse and the other for a keyboard. A special driver would have to be created for the host that would instruct the microcontroller to change configurations. In order to do this, the host's driver would have to be able to detect which peripheral is attached. An easy way to avoid the difficulty of creating a unique driver for the host is to give the microcontroller control over whether it will send the host mouse or keyboard data. Based on the type of data it will send to the PC, the microcontroller can perform a Soft Detach and then re-enumerate as the peripheral of choice.

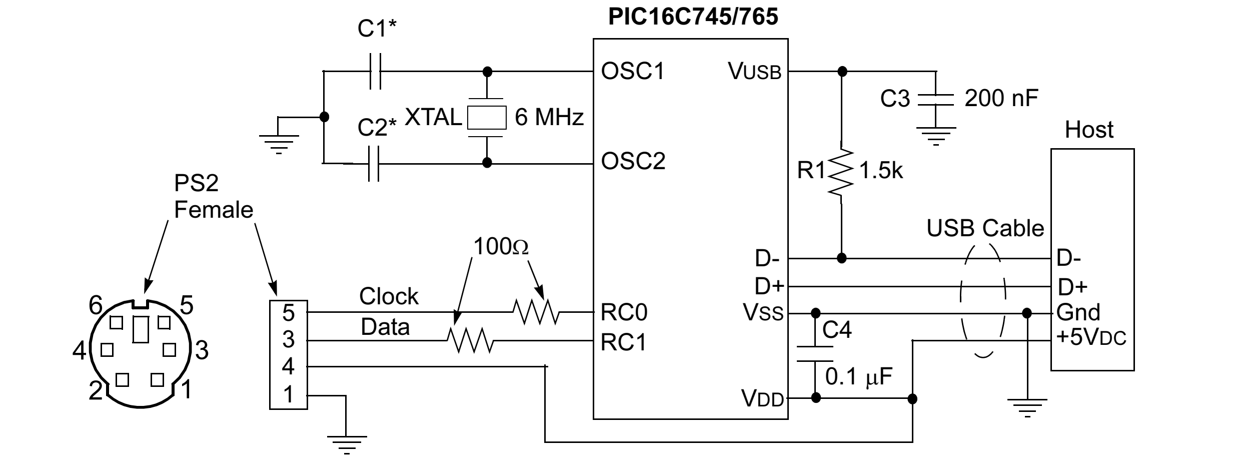
Implementation of Soft Detach

The PICmicro MCU determines the type of device it will enumerate as based on the PS/2 device currently plugged into the translator circuit. For instance, if a PS/2 mouse is plugged into the translator, the PIC16C745/765 will enumerate as a USB mouse. See Figure 1 for the circuit diagram. The translator detects what type of device is attached through the following sequence of events, beginning with a PS/2 device being unplugged.

1. An interrupt is generated when the PS/2 data line goes low.
2. The receive routine is initiated because it is assumed that the data line dropping low is the result of a START bit being sent by the PS/2 device or the PS/2 device has been unplugged.
3. The receive routine times out, indicating that the data line is staying low due to the device being unplugged.
4. The PIC16C745/765 waits for a PS/2 device to be attached.
5. The clock and data lines both go high, indicating a device has been plugged in.
6. The PICmicro MCU firmware asks the PS/2 device to identify itself. (See PS/2 commands and responses in the Appendix.)
7. Based on the PS/2 device's response, the firmware will perform a Soft Detach and re-enumerate as the corresponding USB device.

© 2002 Microchip Technology Inc.

FIGURE 1: TRANSLATOR CIRCUIT DIAGRAM



***Note:** C1 and C2 values selected according to crystal load capacitance.

MULTIPLE DESCRIPTOR SETS

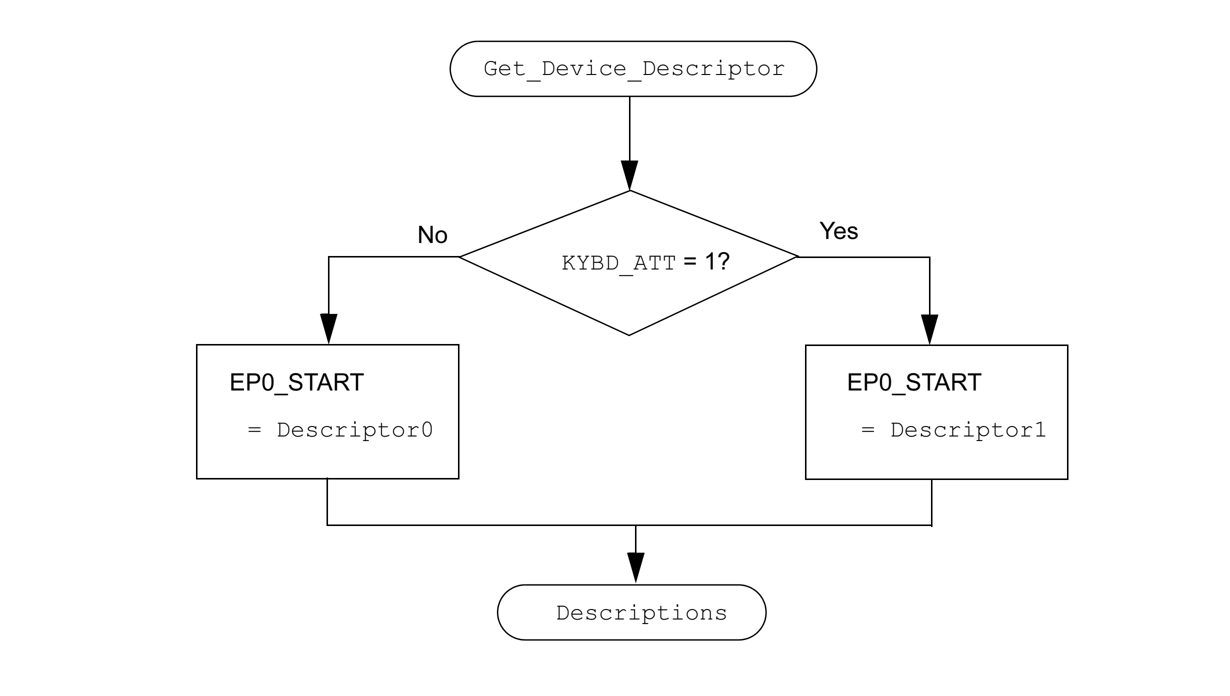
The PICmicro MCU enumerates as either a USB mouse or keyboard by sending one of two sets of descriptors to the host. All Get_Descriptor routines, other than Get_String_Descriptor, use the Descriptions routine to look up a descriptor in the program memory. The following sections will show how the Get_Descriptor routines were modified for multiple descriptor sets. All of these modifications required the use of a flag that indicates what PS/2 device is attached. This flag is KYBD_ATT (keyboard attached). KYBD_ATT is high when the keyboard is attached and low otherwise.

Get_Device_Descriptor, Get_Configuration_Descriptor, and Get_Report_Descriptor

Each of these descriptor routines look up the starting address of the descriptor before calling Descriptions for the first time. Then these routines increment the starting address and call Descriptions repeatedly until every byte of the descriptor has been sent to the host. In order to accommodate for multiple descriptor sets, code was added to each Get_Descriptor routine that ensures the starting address of the corresponding descriptor (mouse or keyboard) is returned based on the status of KYBD_ATT. Figure 2 shows a block diagram of the Get_Device_Descriptor function. Get_Configuration_Descriptor and Get_Report_Descriptor are modified in the same manner as shown in Figure 2.

© 2002 Microchip Technology Inc.

FIGURE 2: GET_DEVICE_DESCRIPTOR ROUTINE



Get_String_Descriptor

No modifications to the Get_String_Descriptor routine are needed. The reason for this is that string descriptors are indexed. In other words, all descriptors (other that string descriptors) have fields where string indexes are specified. For a device descriptor, for instance, the iProduct field may be specified as a 3. This means that String 3 contains product information. This field could have easily been assigned an 8 as long as String 8 contains the product information. Therefore, for multiple descriptor sets, the first set can use strings 0 through 5 and the second set 6 through 10, for instance.

CONCLUSION

The PS/2 to USB translator demonstrates the Soft Detach function by enumerating as either a USB mouse or keyboard depending on which PS/2 device is plugged into it. Soft Detach is a useful feature included in the PIC16C745/765 support firmware because it allows developers to emulate connecting and unconnecting their peripheral while the device is plugged into the host. This characteristic makes it possible for developers to change the entire descriptor set of their device on-the-fly.

MEMORY USAGE

In the PIC16C765, the following memory was used:

• Data Memory:	50 bytes
• Program Memory:	2910 bytes

REFERENCES

1. *USB Specification*, Version 1.1: Chapter 9 (located at www.usb.org)
2. *Device Class Definition for Human Interface Devices* (located at www.usb.org)
3. *HID Usage Tables* (located at www.usb.org)
4. *USB Firmware User's Guide* (located in USB Support Firmware zip file at www.microchip.com)
5. *USB Complete, Second Edition,* Jan Axelson; Lakeview Research, 2001 (www.lvr.com)
6. PS/2 Mouse/Keyboard Protocol, Adam Chapweske,

   http://panda.cs.ndsu.nodak.edu/~achapwes/PICmicro/PS2/ps2.html
7. TB054: *An Introduction to USB Descriptors with a Gameport to USB Gamepad Translator*
8. TB055: *PS/2 to USB Mouse Translator*
9. TB056: *Demonstrating the Set_Report Request with a PS/2 to USB Keyboard Translator Example*
10. TB057: *USB Combination Devices Demonstrated by a Combination Mouse and Gamepad device*

© 2002 Microchip Technology Inc.

APPENDIX A: PS/2 COMMANDS

TABLE 1: HOST TO PS/2 KEYBOARD COMMANDS

HEX Code	Description
ED	Turns on/off LEDs. Keyboard replies with ACK (FA) and waits for another byte to be sent. Next byte sent determines the state of the LEDs (Bits 0-2 correspond to LEDs 1, 2, 3. Bits 3-7 should always be 0).
EE	Echo. Keyboard should respond with Echo (EE).
F0	Set Scan Code Set. Responds with ACK (FA) and waits for another byte to be sent. Next byte sent will be either 01, 02, or 03 (corresponding to scan code sets 1, 2, and 3). If 00 is sent (instead of 01, 02, or 03) keyboard will respond with ACK (FA) followed by the current scan code set (again, 01, 02, or 03).
F2	Get ID. Responds with ACK (FA) followed by an ID (A3, AB). This also enables scanning.
F3	Set repeat rate. Keyboard replies with ACK (FA) and waits for another byte to be sent. Next byte sent will determine the typematic repeat rate for the keyboard. *SEE NOTE BELOW* After this byte is sent, keyboard responds with another ACK (FA).
F4	Enable keyboard. Clears the buffer and starts scanning for data; Replies with ACK (FA).
F5	Disable keyboard. Disables scanning and replies with ACK (FA). Does not affect indicator LEDs.
F6	Restore default values. Does not affect indicator LEDs.
F7	Set all keys typematic. Responds with ACK (FA).
F8	Set all keys make/break. Responds with ACK (FA).
F9	Set all keys make. Responds with AK (FA).
FA	Set all keys typematic/make/break. Responds with ACK (FA).
FB	Set key type typematic.
FC	Set key type make/break.
FD	Set key type make.
FE	Resend. Keyboard responds by retransmitting the last command it sent.
FF	RESET. Resets the keyboard.

TABLE 2: PS/2 KEYBOARD TO HOST COMMANDS

HEX Code	Description
00	Key detection error/keyboard buffer overflow (if set, w or 3 scan codes are enabled).
83,AB	Keyboard ID
AA	Self-test passed
EE	Echo. Sent to Host after receiving “Echo” command from host.
FA	Acknowledge (ACK)
FC	Self-test failed
FE	Resend. Host responds by re-transmitting the last command sent.
FF	Key detection error/keyboard buffer overflow (if set 1 scan codes are enabled)

TABLE 3: PS/2 MOUSE COMMANDS

HEX Code	Description
FF	RESET. This command causes the mouse to enter the RESET mode and do an internal self-reset.
FE	Resend. Any time the mouse receives an invalid command, it returns a resend command to the host system. The host system, in turn, sends this command when it detects any error in any transmission from the mouse. when the mouse receives the resend command, it retransmits the last packet of data sent.
F6	Set Default - This command reinitializes all conditions to the power-on default state.
F5	Disable - This command is used in the Stream mode to stop transmissions initiated by the mouse. The mouse responds to all other commands while disabled. If the mouse is in the Stream mode, it must be disabled before sending it any command that requires a response.
F4	Enable - This command is used in the Stream mode to begin transmission.
F3, XX	Set sampling rate - In the Stream mode, this command sets the sampling rate to the value indicated by byte XX (HEX) /sec.
F2	Read Device Type - This command always receives a response of 0x00 from the mouse.
F0	Set Remote mode - Sets the mouse to Remote mode. Data values are reported on in response to a read data command
EE	Set Wrap mode - Sets the mouse to Wrap mode. The mode remains until 0xFF or 0xEC is received.
EC	Reset Wrap mode - The mouse returns to the previous mode of operation after receiving this command.
EB	Read data - This command requests that all data defined in the data packet format be transmitted. This command is executed in either Remote or Stream mode. This data is transmitted even if there has been no movement since the last report or the switch status is unchanged.
EA	Set Stream mode - this command sets the mouse to Stream mode.
E9	Status request - When this command is issued by the system, the mouse responds with a 3-byte status report, same as Data Report.
E8, XX	Set resolution - The mouse provides 4 resolutions, selected by the second byte of the command.
E7	Set scaling 2:1 - Scaling is used to provide a course/fine tracking response. At the end of a sample interval in the Stream mode, the current X and Y data value are converted to new values. The sign bits are not involved in the conversion. 2:1 scaling is performed only in Stream mode. In response to a read data command, the current value before conversion is sent.
E6	RESET scaling - This command restores scaling to 1:1.

APPENDIX B: SOURCE CODE

Due to the length of the source code for the PS/2 to USB Translator, the source code is available separately. The complete source code is available as a single WinZip archive file, `tb058sc.zip`, which may be downloaded from the Microchip corporate Web site at:

www.microchip.com

Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products 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 PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be 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 product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, KEELOQ, MPLAB, PIC, PICmicro, PICSTART and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

dsPIC, dsPICDEM.net, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICC, PICDEM, PICDEM.net, rfPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

DNV Certification, Inc.
USA

ANSI★RAB
QMS
ACCREDITED

DNV MSC
The Netherlands
Accredited by the RvA

DNV

ISO 9001 / QS-9000
REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001 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

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-4338

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
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, Indiana 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
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

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

ASIA/PACIFIC

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

China - Beijing
Microchip Technology Consulting (Shanghai)
Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu
Microchip Technology Consulting (Shanghai)
Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200 Fax: 86-28-86766599

China - Fuzhou
Microchip Technology Consulting (Shanghai)
Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Shanghai
Microchip Technology Consulting (Shanghai)
Co., Ltd.
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
Microchip Technology Consulting (Shanghai)
Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-82350361 Fax: 86-755-82366086

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

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

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

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Microchip Technology (Barbados) Inc.,
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
Microchip Technology Austria GmbH
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

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

France
Microchip Technology SARL
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
Microchip Technology GmbH
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

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

10/18/02