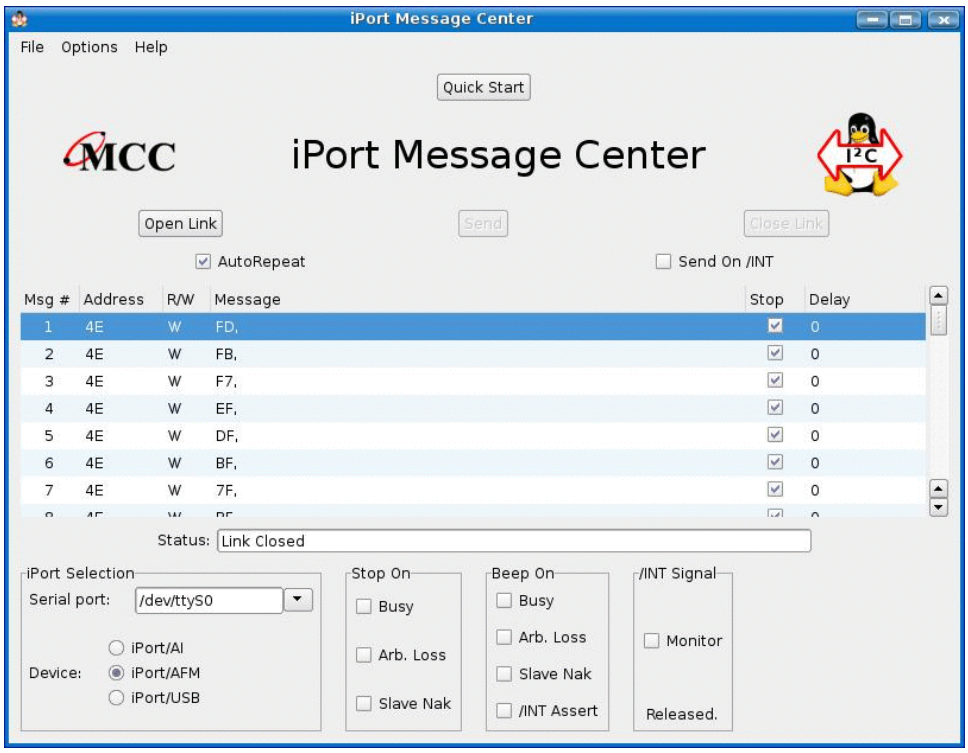
User's Guide

iPort Utility Pack for Linux

I²C Bus Messaging Software



MCC
Micro Computer Control
Small Area Network Specialists

www.mcc-us.com

Introduction

The iPort Message Center for Linux is I²C Bus messaging software that works with MCC I²C Host Adapters with ASCII Text Interface (iPort/USB #MIIC-204, iPort/AFM #MIIC-203, and iPort/AI #MIIC-202). iPort Message Center for Linux allows any Linux based PC with a USB or RS-232 port to become an I²C Master device, transmitting or receiving I²C messages between the PC and one or more I²C devices across an I²C Bus.

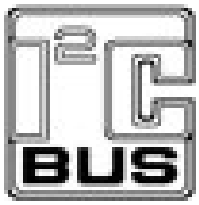
This user's guide describes the installation and operation of the iPort Message Center for Linux.

Are you new to I²C? Want to know more? We suggest you review "What is I²C?" at www.mcc-us.com/I2CBusTechnicalOverview.pdf.

MCC products use Philips components and are licensed to use the I²C Bus.

"Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components of the I²C system, provided the system conforms to the I²C specifications defined by Philips."

I²C is a trademark of Philips Corporation.



23-MAY-07

Copyright© 2007 by Micro Computer Control Corporation. All rights reserved. No part of this publication may be reproduced by any means without the prior written permission of Micro Computer Control Corporation, PO Box 275, Hopewell, New Jersey 08525 USA.

DISCLAIMER: Micro Computer Control Corporation makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Micro Computer Control Corporation reserves the right to revise the product described in this publication and to make changes from time to time in the content hereof without the obligation to notify any person of such revisions or changes.

WARNING - Life Support Applications: MCC products are not designed for use in life support appliances, devices, or systems where the malfunction of the product can reasonably be expected to result in a personal injury.

WARNING - Radio Frequency Emissions: This equipment can radiate levels of radio frequency energy that may cause interference to communications equipment. Operation of this equipment may cause interference with radio, television, or other communications equipment. The user is responsible for correcting such interference at the expense of the user.

WARNING - Electrostatic Discharge (ESD) Precautions: Any damage caused by Electrostatic Discharge (ESD) through inadequate earth grounding is NOT covered under the warranty of this product. See the “Electrostatic (ESD) Precautions” section of this guide for more information.

Printed in the United States of America

Table of Contents

iPort Utility Pack for Linux 1

1. iPort Utility Pack for Linux 2
   iPort Message Center 2

2 System Requirements 3

3 iPort Utility Pack Installation and Removal 3
   Installation 3
   Uninstalling 4

4 iPort Message Center 5
   Message Center Operations 6
      Starting the Message Center 6
      Selecting the Adapter 6
      Selecting the Communications Port 6
      Options Menu 6
      Establish Adapter Communications Link 6
      Entering or Editing I^2C Messages 7
      Set I^2C Address 7
      Set Message Read/Write Direction 7
      Specify Repeated Start Messages 8
      Set Time Delay 8
      Specify Write Data or Read Byte Count 8
      Inserting and Deleting Messages 9
      Saving or Loading Message Lists 9
      Send the Message List 9
      Special Event Handling 10
      Slave Not Acknowledging 10

Appendix A - Linux Serial Help 11

iPort Utility Pack for Linux

V1.0

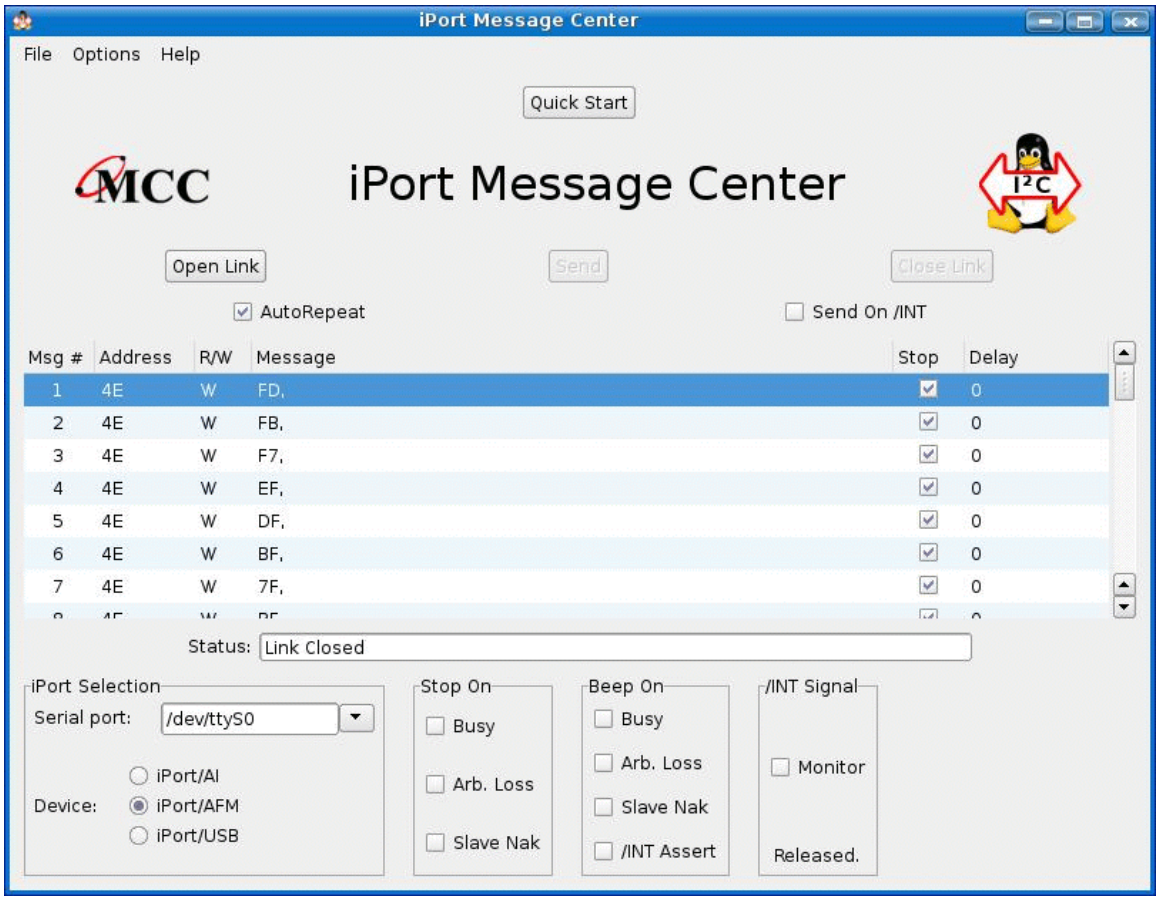
1. iPort Utility Pack for Linux

The iPort Utility Pack for Linux is your express lane to I²C Bus communications. The Utility Pack includes the Linux-based iPort Message Center application that will help you get started sending and receiving I²C Bus messages quickly and easily.

1.1 iPort Message Center

The iPort Message Center, our most popular application, operates with all ASCII text versions of our I²C Bus Host Adapters. With the Message Center, you can create, save, and automatically execute scripts of I²C Bus messages. I²C Bus message activity includes:

- Master Transmit
- Master Receive



2 System Requirements

One of the following MCC I^2C Bus adapters:

- iPort/AI (#MIIC-202)
- iPort/AFM (#MIIC-203)
- iPort/USB (#MIIC-204) (requires iPort/USB Linux driver available at: http://www.mcc-us.com/SoftwareUpgrades-Updates.htm)

32-bit Intel x86 or compatible PC.
Linux kernel V2.4 or higher.
1 free USB or RS-232 Serial Port.
GTK+ V2.0 or higher (included with most major Linux distributions).

3 iPort Utility Pack Installation and Removal

3.1 Installation

To download the application software:

1. Visit MCC's web site (www.mcc-us.com)
2. Click on the Downloads link.
3. Click on the iPort Message Center for Linux link.
4. Click to download either the .TAR.GZ or .ZIP file.
5. Open a command line terminal and navigate to the downloaded file folder.
6. Unpack the downloaded file using the following command:
   For .ZIP: unzip iMsgCtr4Linux100.zip
   For .TAR.GZ: tar-xvvzf iMsgCtr4Linux100.tar.gz

To install the application software, navigate to the unpacked iPortMessageCenter folder and enter the following commands:

If you have "superuser" authorization, enter:
   sudo sh install
   When prompted, enter your user password.

If you do not have "superuser" authorization, enter:
   su
   When prompted, enter the system's root password.
   sh install

3.2 Uninstalling

To uninstall the application software, navigate to the /usr/share/iPortMessageCenter folder and enter the following commands:

If you have “superuser” authorization, enter:

sudo sh uninstall

When prompted, enter your user password.

If you do not have “superuser” authorization, enter:

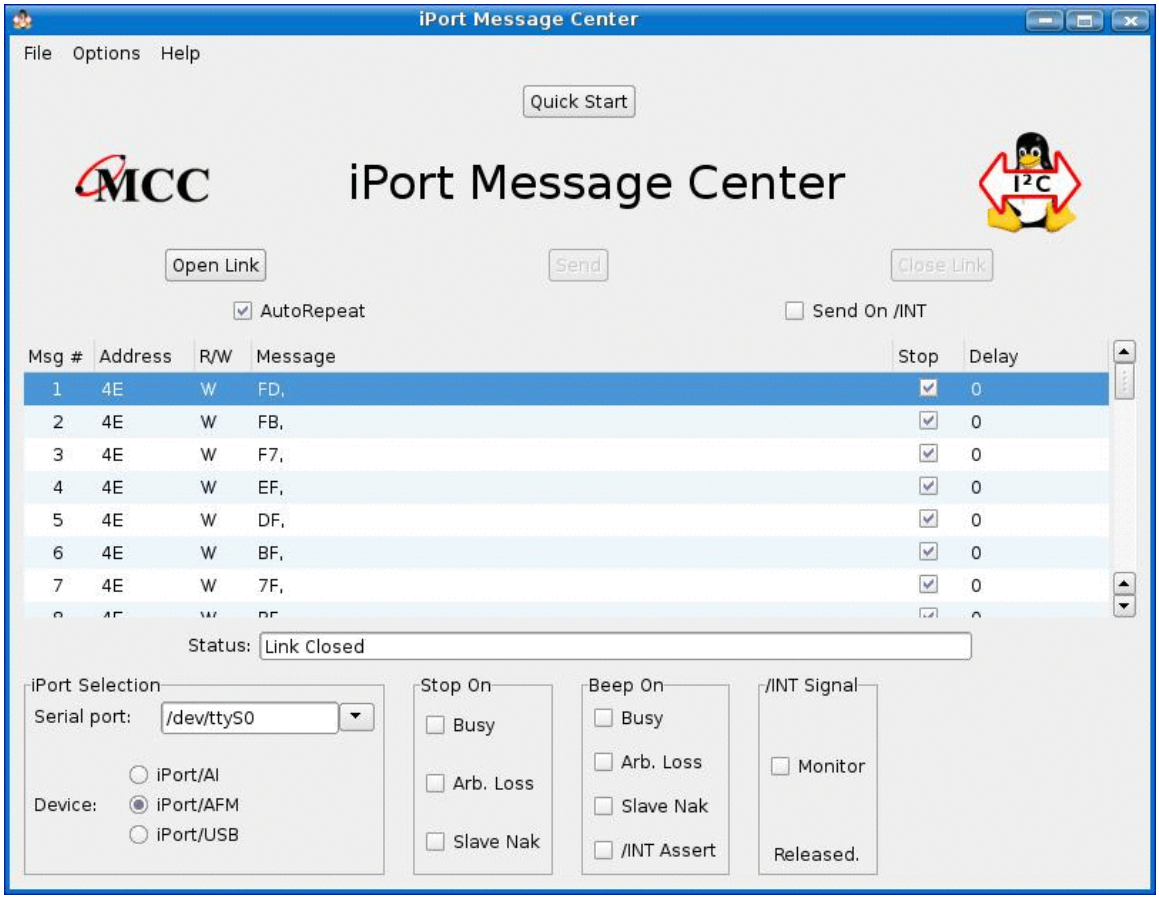
su

When prompted, enter the system’s root password.

sh uninstall

4 iPort Message Center

The iPort Message Center supports I²C Master Transmit and Master Receive activities for all ASCII text based MCC I²C Bus host adapters. With this program you can create, save, and execute scripts of I²C Master messages.



Main Application

The iPort Message Center allows a PC to become an I²C Master transmitter or receiving device, sending I²C messages between the PC and one or more I²C devices across an I²C Bus.

The iPort Message Center is designed to be a simple application for experimenting with I²C messages. It provides methods to:

1. Enter/Edit a list of I²C Master Transmit or Receive Messages.
2. Save and/or Load a list of I²C Master messages to/from disk.
3. Transmit the current list of I²C Master messages, with the option to auto-repeat upon completion, send on INT signal assertion (with INT signal supported adapters only), and beep or stop on special I²C Bus events.

Each I²C message can transfer up to 32 bytes of 8-bit data, with Repeated Start and

Time Delay options.

4.1 Message Center Operations

Communicating with another device on the I^2C Bus is easy. Just install the software as described in Section 3, then following these simple steps:

4.1.1 Starting the Message Center

Open a command line terminal, and enter “iPortMessageCenter” (note: “root” access is not required).

4.1.2 Selecting the Adapter

Select the I^2C adapter you are using by clicking the corresponding adapter Device Select radio button.

4.1.3 Selecting the Communications Port

Use the “Using Com:” control to select the RS-232 communication port connected to the I^2C adapter. (The iPort/USB is supported with the iPort/USB Linux driver available at: http://www.mcc-us.com/SoftwareUpgrades-Updates.htm). NOTE: On some systems (especially 2.4 series kernels), many more serial ports will be listed than actually exist.

4.1.4 Options Menu

Use the Options menu to override default Baud Rate and I^2C Bus Clock rate settings. Default settings and options are adapter dependant.

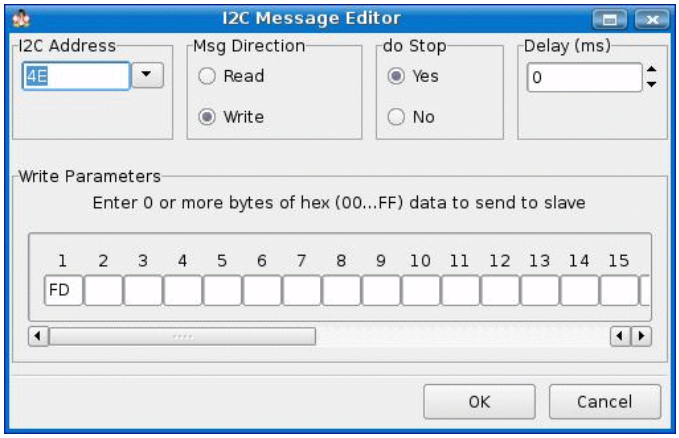
4.1.5 Establish Adapter Communications Link

Establish the communications link to the I^2C adapter by clicking the Open Link button.

The Message Center sets the adapter’s own I^2C Slave address to 0xFE. Once the link has opened successfully, you are an active I^2C node. I^2C messages entered into the message spreadsheet can be transmitted upon request. If the link open is not successful, follow the on-screen directions. Make sure the communications port is working, and is not being used by other software.

4.1.6 Entering or Editing I²C Messages

I²C messages can be entered with the Message Editor, or a previously recorded message list can be loaded from the File menu.



To enter or edit a message, open the "I²C Message Editor" screen by double clicking on a message row in the spreadsheet.

Use the I²C Message Editor to:

1. Set I²C Address.

   The I²C Address is the I²C slave address of the slave device being addressed on the bus. All slave addresses are displayed as even numbers (00...FE), representing the 7 most significant bits of the 8-bit slave address transmitted on the bus (aaaa aaa0).

   The I²C adapter automatically supplies the 8th, least significant, Read/Write bit when it sends the slave address across the bus. For master write operations, the Read/Write bit is always transmitted as a logical 0 (aaaa aaa0). For master read operations, the Read/Write bit is always transmitted as a logical 1 (aaaa aaa1).

   Use the I²C Address control to set the slave address of the slave device you want to address on the bus.

2. Set Message Read/Write Direction.

   As a bus master device, the I²C adapter can write data to, or read data from, any device on the bus. Use the Msg Direction control to specify if the current

message is a master write, or master read, operation. Upon making your selection, additional Write or Read parameters appear.

3. Specify Repeated Start Messages.

   I^2C Bus communications support an operation called Repeated Start. In this operation, a message is sent across the bus beginning with a Start Condition, but without a Stop Condition at the end of the message. The next message sent across the bus begins with a Start Condition, in this case a Repeated Start.

   An I^2C Bus master, that successfully sends a message on the bus, owns the bus until that master sends a message with a terminating Stop Condition. The Repeated Start operation allows the bus master to retain control of the bus while sending one or more messages on the bus. This prevents other bus masters, in a multi-master system, from accessing the bus and interfering with message sequences.

   The Message Center supports Repeated Starts with the doStop control. Sending an I^2C message with doStop enabled will cause the message to be terminated with a Stop Condition. Sending an I^2C message with doStop disabled will cause the message to end without a Stop Condition, allowing the next message to be sent with a Repeated Start.

4. Set Time Delay.

   Message Center supports time delays after the completion of a message. Time delays can be used to synchronize or sequence bus messages with the activity of external devices.

5. Specify Write Data or Read Byte Count.

   Enter the hexadecimal data you want to write to a slave receiver device, or the number of data bytes to read from a slave transmitter. Message Center supports up to 32 bytes of 8-bit data per message.

   NOTE: The data you send may have special meaning to the receiving slave device, but to the Message Center, and the I^2C adapter, message data has no special meaning. Consult your slave device’s data sheet for details.

Click OK to accept the message and enter it into the spreadsheet.

Master Write messages display the message data in the spreadsheet. Master Read messages display 0xFF placeholders in the spreadsheet. Upon execution, actual data received from a slave transmitter replaces the placeholders in the message spreadsheet.

Repeat the above steps for additional messages. The Message Center supports up to 50 messages in a list.

4.1.7 Inserting and Deleting Messages

You can insert a new message between existing messages by clicking once on a message below where you want to insert, then press the “Insert” key on your keyboard. The Message Editor also remembers the last message displayed, so double clicking on a blank spreadsheet row will allow you to copy a message. Delete a message by single clicking on the message row and pressing the “Delete” key on your keyboard.

4.1.8 Saving or Loading Message Lists

Message Center I^2C message lists can be saved to, or loaded from, a disk file. To save the current message list, click File|Save on the menu bar. To open an existing message list, click File|Open List on the menu bar.

Message lists are maintained in ASCII text files (*.IML) that can be edited manually or created with a customer-developed program. See message list files for details.

4.1.9 Send the Message List

An I^2C message list can be sent manually, or automatically in response to an INT signal assertion (with INT signal supported adapters only). To send the list manually, click the Send button on the main application screen. To send the list in response to an INT signal assertion (low), enable the “/INT Signal Monitoring” checkbox, and check the “Send on /INT” checkbox. The list will be sent each time the INT signal is asserted.

The Message Center also supports the repeated sending of a message list. If the Auto Repeat checkbox is checked, a message list will automatically repeat upon completion.

4.1.10 Special Event Handling

The Message Center supports the early termination of a message list, and beep on special events. See the “Stop On” and “Beep On” controls on the main application screen of available options.

4.1.11 Slave Not Acknowledging

If you get a “Slave Not Acknowledging” message in the Status window, this could indicate you have the wrong address in the I^2C Destination Address, or the device is not answering to its address. Some slave devices temporarily stop acknowledging their address. Consult the slave device’s data sheet for details.

Appendix A - Linux Serial Help

Serial Port Help for Linux

This document attempts to solve some of the most common problems with using serial ports under Linux. Every Linux distribution is different, and for distro-specific questions you will need to talk to the distributor, or your system's administrator.

Q: How do I know what serial ports my computer has?

A: In Linux, serial ports are referenced by "device files". To know how to access the serial port, you must know which device file is in charge of that port. To see a listing of available serial ports on your system, enter

```
dmesg | grep tty
```

at the command line. The output will look something like the following:

```
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
```

This indicates that the system has two serial ports, numbered 0 and 1. The device file for each one looks like:

```
/dev/ttySn
```

where "n" is the number of the serial port. So, in the above example, the serial ports would be called "/dev/ttyS0" and "/dev/ttyS1". Note that the leading 0 is not used.

Q: How do I know which serial port my iPort is connected to?

A: Just like in Windows, there's no way to tell which port is connected to your iPort, except by guessing and trial and error. Most modern systems only have one standard serial port, which Linux usually names /dev/ttyS0. Other serial ports, such as unconventional ones on modems and such, are numbered /dev/ttyS1 and higher. Usually you would start with /dev/ttyS0, and if it doesn't seem to work, try one of the other serial ports on your system.

Q: When I attempt to use the sample software, I get errors relating to "write" permissions. What is this?

A: Under Linux, you must have "permission" to access the serial ports. The following two commands will tell you if you have the correct permissions. Again, replace "n" with the desired serial port number:

```
cat /dev/ttySn (press Control-C after a second or two, if nothing happens)
echo 1234 > /dev/ttySn
```

If either of those commands reply with errors about "permissions", then you are not authorized to use the serial ports in their entirety. There are three ways to fix this:

1. Run the software as the "root" user. This is not recommended, as one wrong command as "root" can destroy a system.
2. Ask your administrator to make you a member of the group that is allowed to access the serial port. Usually this group is called "dialout". If you choose this option, note that you will need to log out and back in (or restart the computer) for the changes to take affect.
3. Ask your administrator to make the serial port readable and writable by all users. This can be done with the command "chmod a+rw /dev/ttySn".