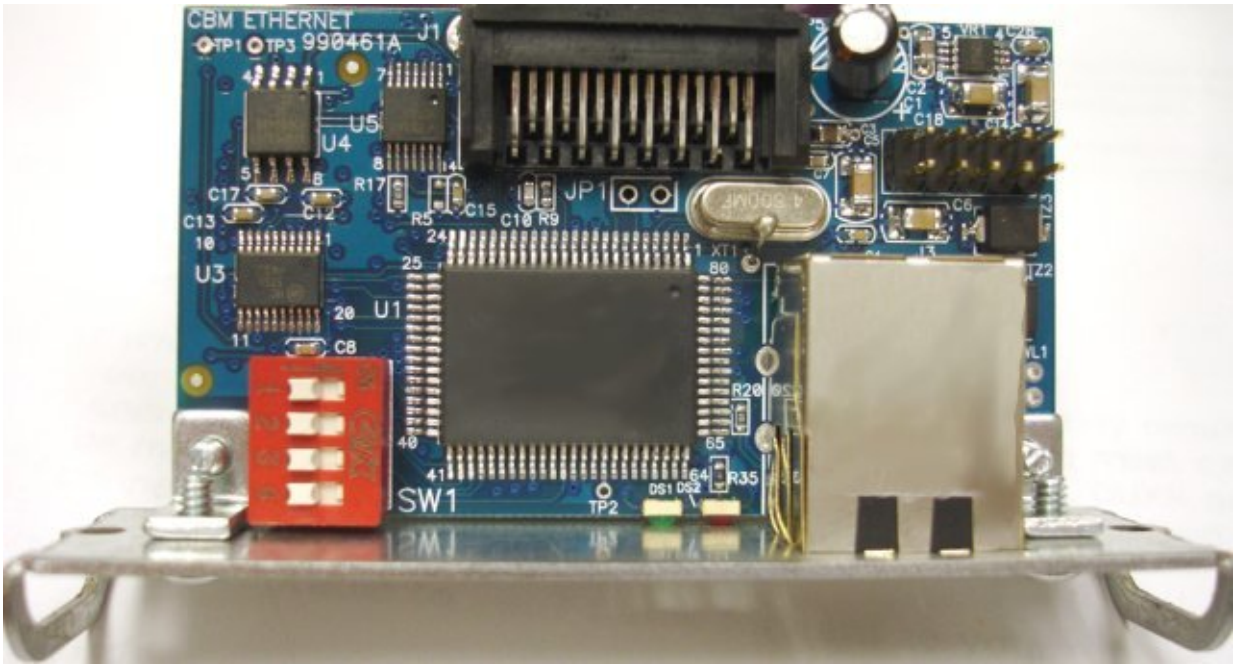
CBM ETHERNET MODULE

PROGRAMMER'S GUIDE



Revision A

04 October 2005

i. Revision History

Revision	***Notes***	***Author***	***Date***
A	Initial revision	Scott	2005/10/04

ii. Related Documents

CBM Ethernet Module Product Overview (CBM America)
CBM Ethernet Utility User's Guide (CBM America)
CBM Ethernet Module Hardware Specification (CBM America)

iii. Preface

Assumptions:

This document assumes the reader has read *CBM Ethernet Module Product Overview* (CyberData Corp.) and *CBM Ethernet Utility User's Guide* (CyberData Corp.) and has familiarized himself with the CBM Ethernet Module in general as well as the details of configuring the CBM Ethernet Module's settings.

Also assumed is a basic knowledge of sockets programming for Windows, Unix, or some other operating system. This document goes over the broad steps for creating and communicating via sockets, but expects the user to understand the underlying mechanisms.

Conventions used in this document:

Commands intended to be typed into a command prompt/shell by the user are indented, boldface and in a slightly smaller font:

```
ping 192.168.1.227
```

Within commands, text enclosed between square brackets is meant to be replaced with whatever value is correct for the user. The below example intends for the user to replace the factory default IP address (enclosed between square brackets) with the correct IP address for his CBM Ethernet Module. Note that the brackets should not be typed when entering the command.

```
telnet [192.168.1.227] 9100
```

Code snippets intended to be used as examples or as a basis for the user's own application are indented and in a slightly smaller font:

```
// Connect and wait for data
if( connect(sockfd, (struct sockaddr *)&serv_addr,  sizeof(serv_addr)) < 0)
{
      perror(NULL);
      printf("Error connecting socket\r\n");

      if( close(sockfd) != 0)
      {
            printf("Close failed\r\n");
            exit(-1);
      }
}
```

Contents

i. Revision History....2
ii. Related Documents....3
iii. Preface....4
Assumptions:....4
Conventions used in this document:....4
I. Overview....6
II. Printing to the CBM Ethernet Module....7
Example 1:....8
Example 2:....9

I. Overview

The CBM Ethernet Module is a low-cost interface designed to bring Ethernet functionality to Citizen CD-S500 and CT-S300 series POS printers.

The CBM Ethernet Module does not provide any emulation or data processing; it simply receives data from the host on an Ethernet socket and passes it to the printer via its serial interface.

Protocols supported by the CBM Ethernet Module include:

- TCP/IP
- UDP
- ARP
- ICMP
- DHCP (client)
- TFTP (server)

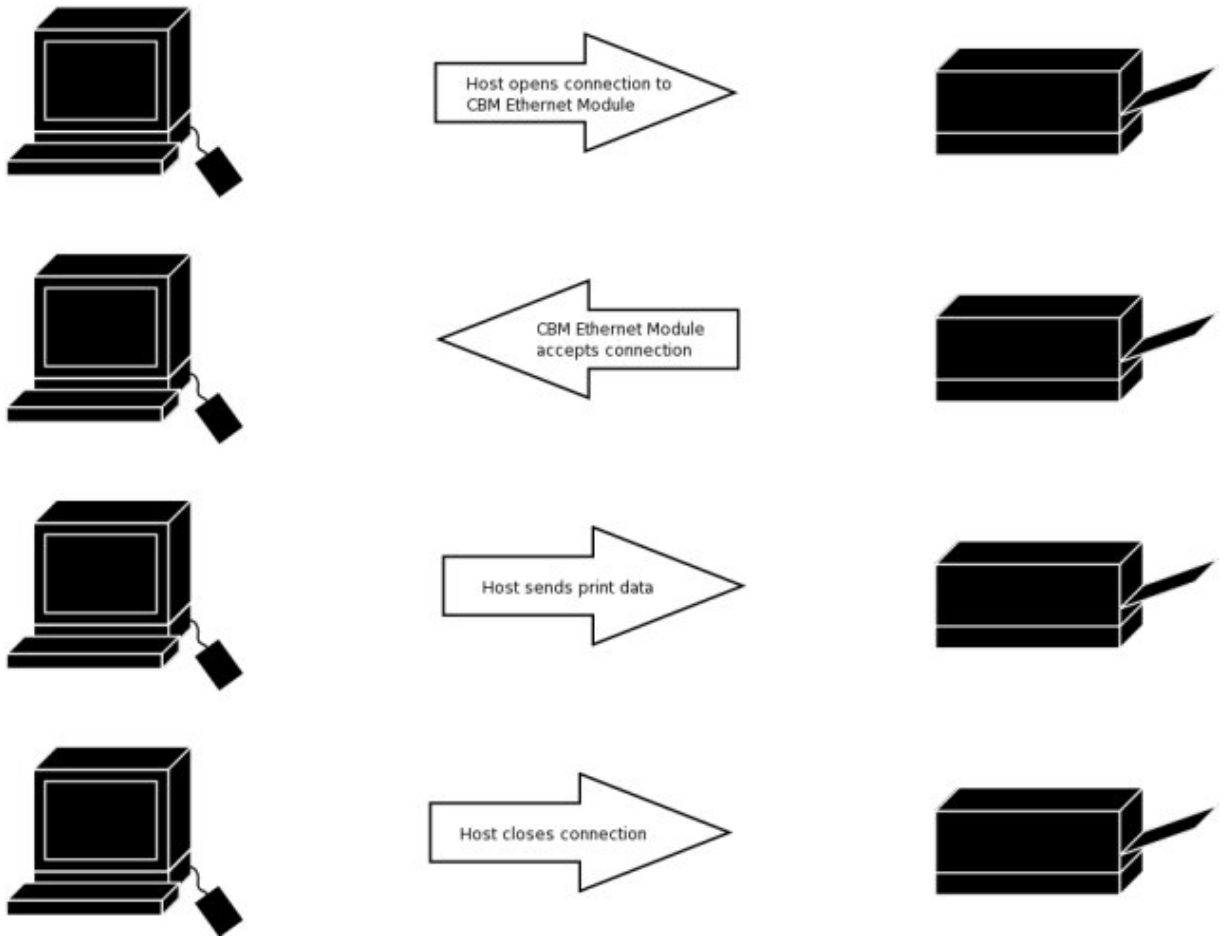
CBM Ethernet Module features include:

- Raw socket printing
- User configurable print port
- User configurable baud rate for serial interface (to printer)
- Remote configuration via CBM Ethernet Utility
- 2 diagnostic LEDs
- Operating system independent

This document is intended to give a general overview on how to communicate with the CBM Ethernet Module and gives some simple examples that may assist in writing your own applications.

II. Printing to the CBM Ethernet Module

Printing to the CBM Ethernet Module is accomplished by creating a TCP socket on the host system and connecting it to the user specified print port of the CBM Ethernet Module (port 9100 by default). The printing flow would look something like the following:



Note that the CBM Ethernet Module can accept only one TCP connection at a time. If the CBM Ethernet Module has an active connection, all subsequent connection attempts will be rejected until the active connection is terminated.

The easiest way – but probably least efficient – to print to the CBM Ethernet Module is to initiate a telnet session and type away. This can be accomplished by bringing up a command prompt (or shell, depending on the operating system) and typing:

```
telnet [192.168.1.227] [9100]
```

When the telnet session is connected, any data typed will be printed on the printer after pressing the **ENTER** key.

Creating a telnet session to the CBM Ethernet Module is fine for diagnostics and testing, but not efficient for a POS environment. Depending on the existing POS application environment, applications may have to be re-written entirely or simply updated to add Ethernet functionality. Below are two examples of possible applications to print to the CBM Ethernet Module.

Example 1:

Cross platform example of a C application using Berkeley sockets:

```
int SendEthernetPrinter(unsigned char * buffer, unsigned int length)
{
	int sockfd = 0;
	struct sockaddr_in serv_addr;

	serv_addr.sin_family = AF_INET;
	serv_addr.sin_addr.s_addr = inet_addr(printer_ip);
	serv_addr.sin_port = htons(atoi(printer_port));

	if( (sockfd = socket(AF_INET, SOCK_STREAM, 0) ) < 0)
	{
		perror(NULL);
		printf("Could not create socket\r\n");
		return(-1);
	}

	// Connect socket
	if(connect(sockfd,(struct sockaddr*)&serv_addr,sizeof(serv_addr))< 0)
	{
		perror(NULL);
		printf("Error connecting socket\r\n");

		if( close(sockfd) != 0)
		{
			printf("Close failed\r\n");
```

```
                        return(-1);
                }

                else
                {
                        // Socket is connected; Send data
                        send(sockfd, buffer, length, 0);
                        return(0)
                }
        }

        return(-1);
}
```

Example 2:

A Unix shell script that pipes data from a file to the CBM Ethernet Module (assumes a Unix script or application that saves the print data as a text file).

```
#! /bin/sh

PRINTER_IP=192.168.1.227
PRINT_PORT=9100
PRINT_FILE= somefile.txt

cat PRINTFILE | netcat -p PRINT_PORT PRINTER_IP
```

These are only basic examples intended to give a brief overview of how to communicate with the CBM Ethernet Module. More detailed tutorials on sockets programming can be found on the world wide web.