C²I² Systems

CCII Systems (Pty) Ltd Registration No. 1990/005058/07

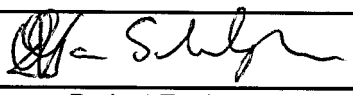
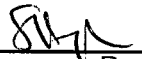
Communications
Computer Intelligence
Integration

User Manual
for the
C²I² Systems Gigabit Ethernet
VxWorks Software Driver

C²I² Systems Document No.	CCII/GE/6-MAN/002
Document Issue	1.0
Issue Date	2007-10-18
Print Date	2007-10-18
File Name	W:\GE\TECH\MAN\CGEMAN02.WPD
Distribution List No	

Document prepared by and for C²I² Systems, Cape Town

Signature Sheet

Name	Signature	Date
Completed by O.G. van Schalkwyk	 Project Engineer GE C²I² Systems	2007-10-18
Accepted by L DELMAR	 Project Manager GE C²I² Systems	2007-10-18
Accepted by S. HAUGH	 Quality Assurance Representative C²I² Systems	2007-10-19

Amendment History

Issue	Description	Date	ECP No.
0.1	First draft.	2007-02-28	-
0.2	Added command line parameter descriptions.	2007-06-07	-
1.0	Procedure added describing the addition of larger Network Stack Memory Pool data clusters.	2007-10-18	CCII/GE/6-ECP/008

Contents

Abbreviations and Acronyms

API	Application Program Interface
BIT	Built-In-Test
bps	bits per second
BSP	Board Support Package
CD	Carrier Detect
CRC	Cyclic Redundancy Check
EEPROM	Electrically Erasable Programable Read Only Memory
FIFO	First In First Out
GE	Gigabit Ethernet
I/O	Input/Output
k	Kilo
LED	Light Emitting Diode
M	Mega
MHz	Megahertz
PC	Personal Computer
PCI	Peripheral Component Interconnect
PMC	PCI Mezzanine Card
POST	Power-On-Self-Test
RAM	Random Access Memory
SBC	Single Board Computer

1. Scope

1.1 Identification

This document is the user manual for the VxWorks software driver for the C²I² Systems Dual Link Gigabit Ethernet Adapter, hereinafter named the GE Adapter.

1.2 System Overview

The C²I² Systems Dual Gigabit Ethernet (GE) adapters attach computers to 10 Mbps, 100 Mbps and 1 Gbps Ethernet networks using Copper or Fibre cabling.

The GE Adapter is currently available in an Air-Cooled PMC, Conduction Cooled PMC and PCI-104 form-factors.

Applicable Part Numbers are:

FP PMC

CCII/GNET/PMC/2P/RJ/FP/COM	UTP Commercial Grade Adapter
CCII/GNET/PMC/2P/RJ/FP/IND	UTP Industrial Grade Adapter
CCII/GNET/PMC/2P/SX/FP/COM	Multimode Fibre Commercial Grade Adapter
CCII/GNET/PMC/2P/SX/FP/IND	Multimode Fibre Industrial Grade Adapter
CCII/GNET/PMC/2P/LX/FP/COM	Single Mode Fibre Commercial Grade Adapter
CCII/GNET/PMC/2P/LX/FP/IND	Single Mode Fibre Industrial Grade Adapter

Conduction-Cooled PMC

CCII/GNET/PMC/2P/BP/CC	UTP, Backplane I/O, Conduction-Cooled Adapter

PCI-104

CCII/GNET/PC104/2P/RJ/COM	UTP Commercial Grade Adapter
CCII/GNET/PC104/2P/RJ/IND	UTP Industrial Grade Adapter
CCII/GNET/PC104/2P/SX/COM	Multimode Fibre Commercial Grade Adapter
CCII/GNET/PC104/2P/SX/IND	Multimode Fibre Industrial Grade Adapter
CCII/GNET/PC104/2P/LX/COM	Single Mode Fibre Commercial Grade Adapter
CCII/GNET/PC104/2P/LX/IND	Single Mode Fibre Industrial Grade Adapter

The software driver binaries are provided with explicit installation instructions.

1.3 Document Overview

This document gives an overview of the Gigabit Ethernet VxWorks software driver installation procedure and its Application Program Interface (API).

2. Applicable and Reference Documents

2.1 Applicable Documents

2.1.1 CCII/GE/6-MAN/001, *Hardware Reference Manual for the C²I² Systems Gigabit Ethernet Adapter*

2.2 Reference Documents

None

3. Software Driver Distribution

The software driver distribution consists of (at least) the following files :

ccGeEnd<arch><.vxworks_version>.a	Host-architecture specific, driver object file : *cc* - CCII Systems (Pty) Ltd *GeEnd* - Enhanced Network Device (END) VxWorks software driver <*arch*> - Host for which the binary is built : • X86 • 18x (Dy4 181/182/183) • Mv5100 (Motorola MVME-5100) <*vxworks_version*> - VxWorks version : • .62 (VxWorks 6.2) • *blank* (VxWorks 5.5.1) e.g. “ccGeEnd18x.62.a” for Gigabit Ethernet software driver built for a DY4 SVME/DMV 181 PowerPC host for VxWorks 6.2.
Readme.txt	General information and installation notes.
Release.txt	Release notes and revision history: Please check this file for information on the latest updates.
sysCcGeEnd.c	PCI initialisation for X86 BSPs. See Appendix A.

4. Installation Procedure

This paragraph describes the installation procedure for the GE VxWorks software driver. (The examples given are for a DY4 SVME/DMV181 PowerPC host.)

4.1 Large UDP packet support

In order to transmit large UDP packets (typically larger than 1992 bytes), the VxWorks "Network Stack Memory Pool Configuration" has to be changed to add larger data clusters. The procedure for doing this differs between VxWorks 5.5 and VxWorks 6.x .

4.1.1 Procedure for VxWorks 5.5 (if building target images using Tornado)

This configuration change can't be performed from the Tornado Project GUI itself. The file {tornado}/target/config/comps/src/net/usrNetLib.c needs to be changed as shown in **Appendix B**.

The example in **Appendix B** adds 10 clusters each of 4096, 8192, 16384, 32768 and 65536 bytes to the Network Stack Memory Pool. This translates to an additional 1,21MB of system memory being allocated to the Network Stack Memory Pool. Depending on the availability of free memory resources, the number of clusters added can be adjusted. A new VxWorks target image has to be built for these changes to take effect.

4.1.2 Procedure for VxWorks 5.5 (if building target images using the command line)

If you are using the command line to build your VxWorks target images, the relevant file to edit will be {tornado}/target/src/config/usrNetwork.c (not usrNetLib.c). The basic procedure and changes though are roughly the same as those indicated in **Appendix B**. A new VxWorks target image has to be built for these changes to take effect.

4.1.3 Procedure for VxWorks 6.x (if building target images using the Workbench IDE)

The Network Stack Memory Configuration can be reconfigured in the Workbench IDE for your VxWorks 6.x target images. Under the Project Navigator tab, double click on Kernel Configuration for the current VxWorks image project. In the Component configuration window, expand and select the "Network Components > Network Core Components > Network Stack Memory Pool Configuration" branch and change the relevant Property Values to include additional (larger than 2048 bytes) clusters. A new VxWorks target image has to be built for these changes to take effect.

4.2 To Build the GE Software Driver into the VxWorks Kernel

Assume the BSP directory is given as : BSP_DIR = /tornado/target/config/dy4181

- Copy ccGeEnd18x.a to your $(BSP_DIR)/lib directory as ccGeEnd.a.
- In the Builds section of the Project Workspace, change the Kernel properties to include the ccGeEnd.a library file in the Macros LIBs option.
- Rebuild all VxWorks images.

4.3 To Load the Software Driver Separately

Note : This step is not required if the software driver was built into the BSP.

If the software driver is not built into the BSP, a user can load it separately :

- Copy ccGeEnd18x.a to your present working directory as ccGeEnd.a.
- From the VxWorks shell type :
  - ld < ccGeEnd.a

5. Using the GE Software Driver

5.1 DualNet and RLMT Modes

The two ports of the GE dual port adapter may be used either to provide two independent communication channels (DualNet mode) or as a single dual redundant channel (RLMT mode).

In DualNet mode, each channel is assigned is own IP address.

In RLMT mode, the secondary port becomes a “hot standby” in the event of failure of the primary port. Only one IP address is assigned as the hardware will determine over which physical channel the data is routed.

5.2 Loading and Starting the Driver in DualNet Mode

To start the driver in DualNet mode :

```
muxDevStart(muxDevLoad(0,geLoad,"",0,0))
muxDevStart(muxDevLoad(1,geLoad,"",0,0))
```

This will create two devices, “ccge0" and “ccge1".

5.3 Loading and Starting the Driver in RLMT Mode

To start the driver in RLMT mode :

```
muxDevStart(muxDevLoad(0,geLoad,"RlmtMode=CheckLinkState",0,0))
```

5.4 Command Line Parameters

The general form of the GE driver load command is as follows :

muxDevLoad(*device*,geLoad,*"parameter-1:parameter-2:parameter-3...*",0,0)

where *device* is a unique instance number starting from 0, and *parameter-n* is as described in the following table :

Parameter :	**Description :**	**Permitted Values :**	**Default Value:**
Speed_A Speed_B	This parameter is used to set the speed capabilities. It is only valid for copper adapters. Usually, the speed is negotiated between the two ports during link establishment. If this fails, a port can be forced to a specific setting with this parameter.	10 100 1000 Auto	Auto
AutoNeg_A AutoNeg_B	The "Sense"-mode automatically detects whether the link partner supports auto-negotiation or not.	On Off Sense	On
DupCap_A DupCap_B	This parameters is only relevant if auto-negotiation for this port is not set to "Sense". If auto-negotiation is set to "On", all three values are possible. If it is set to "Off", only "Full" and "Half" are allowed. This parameter is useful if your link partner does not support all possible combinations.	Half Full Both	Both
FlowCtrl_A FlowCtrl_B	This parameter can be used to set the flow control capabilities the port reports during auto-negotiation. It can be set for each port individually. • **Sym** = Symmetric: both link partners are allowed to send PAUSE frames • **SymOrRem** = SymmetricOrRemote: both or only remote partner are allowed to send PAUSE frames • **LocSend** = LocalSend: only local link partner is allowed to send PAUSE frames • **None** = no link partner is allowed to send PAUSE frames	Sym SymOrRem LocSend None	SymOrRem

<table>
<tr><th>Parameter :</th><th>Description :</th><th>Permitted Values :</th><th>Default Value:</th></tr>
<tr><td>Role_A
Role_B</td><td>This parameter is only valid for the copper adapters. For two 1000Base-T ports to communicate, one must take the role of the master (providing timing information), while the other must be the slave. Usually, this is negotiated between the two ports during link establishment. If this fails, a port can be forced to a specific setting with this parameter.</td><td>Auto
Master
Slave</td><td>Auto</td></tr>
<tr><td>ConType</td><td>The parameter 'ConType' is a combination of all five per-port parameters within one single parameter. This simplifies the configuration of both ports of an adapter card. The different values of this variable reflect the most meaningful combinations of port parameters.

The following table shows the values of 'ConType' and the corresponding combinations of the per-port parameters:

<table>
<tr><th>ConType</th><th>DupCap</th><th>AutoNeg</th><th>FlowCtrl</th><th>Role</th><th>Speed</th></tr>
<tr><td>Auto</td><td>Both</td><td>On</td><td>SymOrRem</td><td>Auto Auto</td><td></td></tr>
<tr><td>1000FD</td><td>Full</td><td>Off</td><td>None</td><td>Auto (ignored)</td><td>1000</td></tr>
<tr><td>100FD</td><td>Full</td><td>Off</td><td>None</td><td>Auto (ignored)</td><td>100</td></tr>
<tr><td>100HD</td><td>Half</td><td>Off</td><td>None</td><td>Auto (ignored)</td><td>100</td></tr>
<tr><td>10FD</td><td>Full</td><td>Off</td><td>None</td><td>Auto (ignored)</td><td>10</td></tr>
<tr><td>10HD</td><td>Half</td><td>Off</td><td>None</td><td>Auto (ignored)</td><td>10</td></tr>
</table>

Stating any other port parameter together with this 'ConType' variable will result in a merged configuration of those settings. This due to the fact, that the per-port parameters (e.g. Speed_?) have a higher priority than the combined variable 'ConType'.

<u>NOTE:</u> This parameter is always used on both ports of the adapter card.</td><td>Auto
1000FD
100FD
100HD
10FD
10HD</td><td>Auto</td></tr>
<tr><td>PrefPort</td><td>This is used to force the preferred port to A or B (on dual-port network adapters). The preferred port is the one that is used if both are detected as fully functional.</td><td>A
B</td><td>A</td></tr>
<tr><td>RlmtMode</td><td>RLMT monitors the status of the port. If the link of the active port fails, RLMT switches immediately to the standby link. The virtual link is maintained as long as at least one 'physical' link is up.

• CheckLinkState - Check link state only: RLMT uses the link state reported by the adapter hardware for each individual port to determine whether a port can be used for all network traffic or not.
• CheckLocalPort - In this mode, RLMT monitors the network path between the two ports of an adapter by regularly exchanging packets between them. This mode requires a network configuration in which the two ports are able to "see" each other (i.e. there must not be any router between the ports).
• CheckSeg - Check local port and segmentation: This mode supports the same functions as the CheckLocalPort mode and additionally checks network segmentation between the ports. Therefore, this mode is only to be used if Gigabit Ethernet switches are installed on the network that have been configured to use the Spanning Tree protocol.
• DualNet - In this mode, ports A and B are used as separate devices. If you have a dual port adapter, port A will be configured as eth0 and port B as eth1. Both ports can be used independently with distinct IP addresses. The preferred port setting is not used. RLMT is turned off.

<u>NOTE:</u> RLMT modes CLP and CLPSS are designed to operate in configurations where a network path between the ports on one adapter exists. Moreover, they are not designed to work where adapters are connected back-to-back.</td><td>CheckLinkState
CheckLocalPort
CheckSeg
DualNet</td><td>DualNet</td></tr>
</table>

Parameter :	Description :	Permitted Values :	Default Value:
IntsPerSec	This parameter is only used, if either static or dynamic interrupt moderation is used on a network adapter card. Using this parameter if no moderation is applied, will lead to no action performed. This parameter determines the length of any interrupt moderation interval. Assuming that static interrupt moderation is to be used, an 'IntsPerSec' parameter value of 2000 will lead to an interrupt moderation interval of 500 microseconds. **NOTE:** The duration of the moderation interval is to be chosen with care. At first glance, selecting a very long duration (e.g. only 100 interrupts per second) seems to be meaningful, but the increase of packet-processing delay is tremendous. On the other hand, selecting a very short moderation time might compensate the use of any moderation being applied.	30 ... 40 000 (interrupts per second)	2 000
Moderation	Interrupt moderation is employed to limit the maximum number of interrupts the driver has to serve. That is, one or more interrupts (which indicate any transmit or receive packet to be processed) are queued until the driver processes them. When queued interrupts are to be served, is determined by the 'IntsPerSec' parameter, which is explained later below. • **None** - No interrupt moderation is applied on the adapter card. Therefore, each transmit or receive interrupt is served immediately as soon as it appears on the interrupt line of the adapter card. • **Static** - Interrupt moderation is applied on the adapter card. All transmit and receive interrupts are queued until a complete moderation interval ends. If such a moderation interval ends, all queued interrupts are processed in one big bunch without any delay. The term 'static' reflects the fact, that interrupt moderation is always enabled, regardless how much network load is currently passing via a particular interface. In addition, the duration of the moderation interval has a fixed length that never changes while the driver is operational. • **Dynamic** - Interrupt moderation might be applied on the adapter card, depending on the load of the system. If the driver detects that the system load is too high, the driver tries to shield the system against too much network load by enabling interrupt moderation. If - at a later time - the CPU utilizaton decreases again (or if the network load is negligible) the interrupt moderation will automatically be disabled. Interrupt moderation should be used when the driver has to handle one or more interfaces with a high network load, which - as a consequence - leads also to a high CPU utilization. When moderation is applied in such high network load situations, CPU load might be reduced by 20-30%. **NOTE:** The drawback of using interrupt moderation is an increase of the round-trip-time (RTT), due to the queueing and serving of interrupts at dedicated moderation times.	None Static Dynamic	Dynamic
LowLatency	This is used to reduce the packet latency time of the adapter. Setting the LowLatency parameter to 'On' forces the adapter to pass any received packet immediately to upper network layers and to send out any transmit packet as fast as possible. **NOTE 1:** The system load increases if LowLatency is set to 'On' and a lot of data packets are transmitted and received. **NOTE 2:** This parameter is only used on adapters which are based on PCI Express compatible chipsets.	On Off	Off
BroadcastPrio	This parameter specifies whether received broadcast packets have the highest priority for the port switch decision ("Off") or not ("On").	On Off	Off
RlmtMinToVal	Minimum timeout value for RLMT (in µs).		30 000

Parameter :	Description :	Permitted Values :	Default Value:
RlmtDefToVal	Minimum timeout value for RLMT (in µs).		30 000
RlmtPortdownTimVal	RLMT Port Down Timer (in µs)		90 000
RlmtPortstartTimVal	RLMT Port Start Timer (in µs)		50 000
RlmtPortupTimVal	RLMT Port Up Timer (in µs)		2 500 000
RlmtSegToVal	RLMT Network Segmentation Reporting Interval (in µs)		900 000 000

Note that if DualNet mode is used, the parameters for both ports must be provided in the first call to muxDevLoad. Parameters provided in the second call will be ignored.

6. Contact Details

6.1 Contact Person

Direct all correspondence and / or support queries to the Project Manager (HSS8) at C²I² Systems.

6.2 Physical Address

C²I² Systems (Pty) Ltd
Unit 3, Rosmead Place
Rosmead Centre
67 Rosmead Avenue
Kenilworth
Cape Town
7708
South Africa

6.3 Postal Address

C²I² Systems (Pty) Ltd
P.O. Box 171
Rondebosch
Cape Town
7701
South Africa

6.4 Voice and Electronic Contacts

Tel : (+27) (0)21 683 5490
Fax : (+27) (0)21 683 5435
Email : info.ccii@ccii.co.za
Email : support@ccii.co.za
URL : http://www.ccii.co.za/

6.5 Product Support

Support on C²I² Systems' products is available telephonically between Monday and Friday from 09:00 to 17:00 CAT. Central African Time (CAT = GMT + 2).

Email support is available at support@ccii.co.za

Appendix A

Making Changes to sysNet.c for X86

On X86 platforms, the following amendment must be made to the BSP :

Copy the file sysCcGeEnd.c to your BSP configuration directory (target/config/BSPName).

Make the following two amendments to the file sysNet.c in the same directory :

Amendment 1 :

```
#ifdef  INCLUDE_END
#   include "sysDec21x40End.c"          /* dec21x40End support routines */
#   include "sysEl3c90xEnd.c"           /* el3c90xEnd support routines  */
#   include "sysElt3c509End.c"          /* elt3c509End support routines */
#   include "sysFei82557End.c"          /* fei82557End support routines */
#   include "sysGei82543End.c"          /* gei82543End support routines */
#   include "sysLn97xEnd.c"             /* ln97xEnd support routines    */
#   include "sysNe2000End.c"            /* ne2000End support routines   */
#   include "sysUltraEnd.c"             /* ultraEnd support routines    */
#   include "sysCcGeEnd.c"              /* CCII GE END driver support routines */
#endif  /* INCLUDE_END */
```

Amendment 2 :

```
LOCAL VEND_ID_DESC vendorIdEnet [] =
    {
    #if defined(INCLUDE_DEC21X40_END)
    {DEC_PCI_VENDOR_ID,      sysDec21x40PciInit},
    #endif /* INCLUDE_DEC21X40_END */

    #if defined(INCLUDE_LN_97X_END)
    {AMD_PCI_VENDOR_ID,      sysLan97xPciInit},
    #endif /* INCLUDE_LN_97X_END */

    #if defined(INCLUDE_EL_3C90X_END)
    {THREECOM_PCI_VENDOR_ID, sysEl3c90xPciInit},
    #endif /* INCLUDE_EL_3C90X_END */

    #if defined(INCLUDE_GEI8254X_END) || defined(INCLUDE_GEI_HEND)
    {INTEL_PCI_VENDOR_ID,    sys543PciInit},
    #endif /* INCLUDE_GEI8254X_END */

    #if defined(INCLUDE_FEI_END)
    {INTEL_PCI_VENDOR_ID,    sys557PciInit},
    #endif /* INCLUDE_FEI_END */

    {0x1148, sysCcGePciInit},

    {0xffffffff, NULL}        /* last entry */
    };
```

Now remake your VxWorks image.

Appendix B

Making changes to usrNetLib.c for large UDP packets

```
/* OVS: Added these lines to add larger Network Data Pool clusters to the VxWorks image
*/
/* OVS:  To restore file to original config, simply delete all lines marked "OVS"  */

/*  OVS */  #undef NUM_CL_BLKS
/*   OVS */   #define NUM_CL_BLKS (NUM_64 + NUM_128 + NUM_256 + NUM_512 + NUM_1024 +
NUM_2048 + NUM_4096 + NUM_8192 + NUM_16384 + NUM_32768 + NUM_65536)
/*  OVS */  #undef  NUM_64
/*  OVS */  #define NUM_64 100
/*  OVS */  #undef  NUM_128
/*  OVS */  #define NUM_128 100
/*  OVS */  #undef  NUM_256
/*  OVS */  #define NUM_256 40
/*  OVS */  #undef  NUM_512
/*  OVS */  #define NUM_512 40
/*  OVS */  #undef  NUM_1024
/*  OVS */  #define NUM_1024 25
/*  OVS */  #undef  NUM_2048
/*  OVS */  #define NUM_2048 25
/*  OVS */  #define NUM_4096 10
/*  OVS */  #define NUM_8192 10
/*  OVS */  #define NUM_16384 10
/*  OVS */  #define NUM_32768 10
/*  OVS */  #define NUM_65536 10

/*  OVS */  #undef  NUM_NET_MBLKS
/*  OVS */  #define NUM_NET_MBLKS (2* NUM_CL_BLKS)

CL_DESC clDescTbl [] =
    {
    /*
    clusterSize         num             memArea         memSize
    -----------         ----            -------         -------
    */
    {64,                NUM_64,         NULL,           0},
    {128,               NUM_128,        NULL,           0},
    {256,               NUM_256,        NULL,           0},
    {512,               NUM_512,        NULL,           0},
    {1024,              NUM_1024,       NULL,           0},
    {2048,              NUM_2048,       NULL,           0},
/*  OVS */      {4096,          NUM_4096,       NULL,           0},
/*  OVS */      {8192,          NUM_8192,       NULL,           0},
/*  OVS */      {16384,         NUM_16384,      NULL,           0},
/*  OVS */      {32768,         NUM_32768,      NULL,           0},
/*  OVS */      {65536,         NUM_65536,      NULL,           0}
    };
```