

CobraNet LE

Software API

Date 11/21/2008

Revision 001

Contents

1 - Overview	4
2 - Management Interface Variable Reference.....	6
2.1 - Legend	6
2.2 - Data Types	6
2.3 - MIB-II Variables	7
2.3.1 - System	7
2.3.2 - Interface	10
2.3.3 - IP	17
2.3.4 - UDP	24
2.3.5 - SNMP	26
2.4 - CobraNet Variables	34
2.4.1 - Firmware	34
2.4.2 - Hardware Identification	37
2.4.3 - Flash	38
2.4.4 - Errors	40
2.4.5 - Conductor	42
2.4.6 - Conductor Information	44
2.4.7 - Packet Bridge	46
2.4.8 - Serial Bridge	46
2.4.9 - Interrupt Control	48
2.4.10 - Audio	48
2.4.11 - Receivers	48
2.4.12 - Transmitters	51
2.4.13 - Synchronization	54
2.4.14 - SNMP Monitor	56
2.4.15 - MI Monitor	56
2.4.16 - IP Monitor	57
2.4.17 - IF Monitor	58
2.5 - DSP Extensions	60
2.6 - User Extensions	60
2.7 - System Extensions	64
3 - SHMI message protocol	65
3.1 - Terminology	65
3.2 - System Description	66
3.3 - Serial Format	66
3.3.1 - Message Format	66
3.3.2 - Read Messages	67
3.3.2.1 - Read Message Response	67
3.3.3 - Write Messages	68
3.3.3.1 - Write Message Response	68
3.3.4 - Baud Rate Change Message	69
3.3.5 - <i>Nack</i> Response Descriptions	70
3.4 - Commands & Responses	71
3.4.1 - Legend	72
3.4.2 - sysDescr	72
3.4.3 - sysContact	73
3.4.4 - sysName	73
3.4.5 - sysLocation	73
3.4.6 - ifPhysAddress	74
3.4.7 - flashPersistEnable	74
3.4.8 - errorCode	74
3.4.9 - errorCount	74
3.4.10 - modeRateControl	75
3.4.11 - conductorStatus	75
3.4.12 - serialFormat	75
3.4.13 - serialBaud	76
3.4.14 - serialRxMAC	76
3.4.15 - serialTxMAC	76
3.4.16 - rxSubFormat	77
3.4.17 - rxBundle	77
3.4.18 - rxSubMap	78
3.4.19 - txBundle	78

3.4.20 - txSubCount	78
3.4.21 - txUnicastMode.....	79
3.4.22 - txMaxUnicast.....	79
3.4.23 - txSubMap	80
3.4.24 - txSubFormat.....	80
3.4.25 - ipMonCurrentIP	81
3.4.26 - stdUserVersionMajor	81
3.4.27 - stdUserVersionMinor	81
3.4.28 - stdUserString.....	82
3.4.29 - stdUserInteger.....	82
3.4.30 - sysMAC	82
4 - Error Code Reference.....	83
4.1 - Legend	83
4.2 - Error Code Interpretation	83
4.3 - Error Codes Listing.....	84
5 - Notice of Open Source Software Components used in the CobraNet LE firmware.....	87
Bootloader code.....	87
AES Encryption Algorithm	87
Operating System.....	88
TCP/IP stack.....	88
SNMP	89
APPENDIX A - Introduction to CobraNet.....	A-1
APPENDIX B - Reference Documents	B-1

1 - Terminology

The following table describes abbreviations, acronyms, and technical terms used.

Term	Meaning
Ack	Positive Acknowledgement
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
HMI	Host Management Interface
SHMI	Serial Host Management Interface
IP	Internet Protocol
bps	Bits Per Second
MAC	Media Access Controller
MI	Management Interface
Nack	Negative Acknowledgement
SCI	Serial Communications Interface
SSI	Synchronous Serial Interface

2 - Overview

The CobraNet LE uses a subset of the full CobraNet implementation. It implements all the major features of a standard CobraNet node. Some features have also been added to support specific new features of the CobraNet LE. A number of features have been modified and some features and variables are not supported. Below is a complete list of the changes.

Additions

- Serial setting of device MAC address.
- Major and minor user firmware version parameters.
- Four 60-character user string parameters (2 are persistent).
- 64 user integer parameters (32 are persistent).
- GPIO port control and data parameters.
- ADC port control and data parameters.

Modifications

- Supports only 5-1/3 ms latency at 48 kHz.
- ipDefaultTTL is set to 256 by default not 128.
- Reduced settings for txSubFormat and rxSubFormat (see Table 1 below).
- Serial Bridge has a set format (8 data, no parity, 1 stop) and only supports RS232.
- Serial Bridge format maximum baud rate is 57600.
- Flash data (firmware updates) may only be done via TFTP.
- The *firmwareHardwarePlatform* parameter has additional results when read to identify the CobraNet LE.
- The *firmwareHardwareVersion* parameter has additional results when read to identify the CobraNet LE.
- errorCode parameter shows last 4 errors, each as 8-bit value.
- modeRateControl is read-only and always returns 0x600.
- conductorCycleRate is read-only.
- conductorPriority is read-only and always returns 0.
- Usable transmit bundles reduced to 2.
- Usable receive bundles reduced to 2.
- syncConductorClock, syncPerformerClock set to read-only and always report 0.
- Unrecoverable faults halt the CobraNet LE. All audio and Control data delivery is suspended.
- sysDescr reports "Attero Tech CobraNet LE version 2.1.3 LM3S6918" by default.
- sysMAC and SysDesc can be updated via SHMI.

Removals

- SNMP "get-next" calls not supported.
- No support for output channel duplication.
- No external watchdog signal. The watchdog is dealt with internally.
- No support for POST LED fault reporting.
- No support for packet bridge.
- No support for metering.
- No support for the buddy system.
- No support for being a conductor.
- No support for interrupt control.
- No support for audio parameters such as *audioAllowedChannels*.
- No support for redundant Ethernet ports.
- No support for DSP extensions.
- No support for private bundles.
- No support for *snmpMonitor* parameter.
- snmpSilentDrops and snmpProxyDrops parameters not implemented.
- *firmwareFreeCycles* parameter not implemented.
- *flashPersistSize*, *flashTacknowledge*, *flashTrequest*, *flashTAddress*, *flashTLength*, and *flashTDirection* parameters not implemented.
- *errorPOSRRResults*, *errorIndicators*, and *errorDisplay* parameters not implemented.
- *serialTxPriority* and *serialTxBundle* not implemented.
- *rxDelay*, *rxMinDelay*, *rxSourceMAC*, *rxPriority*, and *rxBuddyExclude* parameters not implemented.
- *txPosition*, *txDestinationMAC*, *txPriority*, and *txBuddyExclude* parameters not implemented.
- *syncClockTrim* and *syncBuddyLinkControl* parameters not implemented.
- *miMonHMIMode* parameter not implemented.
- *ifmSwitchMode* parameter not implemented.

Audio Format Value (decimal)	Resolution	Sample Rate	Latency
0	No Signal		
278,528 (0x44000)	16 bit	48 kHz	5 1/3 ms
344,064 (0x54000)	20 bit	48 kHz	5 1/3 ms
409,600 (0x66000)	24 bit	48 kHz	5 1/3 ms

Table 1 - CobraNet LE Subformat Options

The Audio I/O map is fixed in CobraNet LE and cannot be modified and the LE does not support the *audioMap* variable. Audio inputs starting from the channel labeled 1 will be assigned starting at routing channel 1. Audio outputs starting from the channel labeled 1 will be assigned starting at routing channel 33.

It is recommended that all front panel interfaces allow selection of multicast bundles in the range 1-255 and unicast bundles in the range 256-65279. CobraNet LE does not support private bundle assignments (bundle numbers 65280 -65535).

3 – Management Interface Variable Reference

CobraNet interfaces are configured and monitored by reading and writing Management Interface variables. MI variables may be accessed directly via a processor attached to the Serial Host Management Interface (SHMI) or via the network using SNMP. The method of using the SHMI is similar for all Attero Tech CobraNet interfaces, but may require specific semantics and may have different word sizes depending on the actual implementation. The SHMI interface is described in more detail in section 4 of this document. Following are detailed descriptions of the size, contents, and effects of the MI variables as well as their SHMI addresses and SNMP Object Identifier numbers. All MI variables can be accessed via the SHMI but some variable properties render them inappropriate for access via SNMP. These exceptions are noted in the variable descriptions where applicable.

3.1 – Legend

Name	Name of variable as seen in CobraNet Management Information Base (MIB).
Description	Description of the variable including allowed values and usage discussion.
Host Address	SHMI addresses are used to access variables via the host port. SHMI addresses have a 24-bit range.
SNMP Object ID (OID)	The Object Identifier is the numeric name assigned to a variable according to the SNMP protocol.
Size	Size is indicated for varying-length data types such as DisplayString and OID. <i>Size</i> is not indicated for fixed types whose size is implied by their data type.
Count	Number of entries for array or buffer-type variables. Absence of a Count specification implies a single instance variable, and thus a Count of 1.
Type	Data type of the variable. The options and format of data types are described below in detail.
Attributes	Read-only variables can only be read and can not be modified. Read/Write variables can be read and written. Read/Write - Persistent variables can be read and written. If the persistence feature is enabled, values of these variables will automatically be written to flash for recall at startup.
Default	Value assigned to the variable at startup when persistence is disabled. The values of some read-only variables reflect system conditions and thus may not have a default value.
Implemented Version	Firmware version in which the variable was first introduced. Unless otherwise noted in this field, one can assume variables will be available in the version indicated and all subsequent versions.

3.2 – Data Types

DisplayString	A <i>DisplayString</i> is an ASCII string comprised entirely of printable characters.
OID	An SNMP object identifier is the numeric name of an SNMP variable. OIDs are also used for other purposes including system-unique identifiers.
IpAddress	An <i>IpAddress</i> is a 32-bit internet protocol (IP) address.
PhysAddress	A 48-bit Ethernet media access control (MAC) address.
TimeTicks	<i>TimeTicks</i> is an integer encoding for time durations in units of 100ths of a second. SNMP: <i>TimeTicks</i> is reported as a 32-bit integer in units of 100ths of a second. For example, a reported value of <i>1000</i> indicates a 10-second timer reading. As seen through SNMP, <i>TimeTicks</i> variables roll over after 2^{32} 100ths of a second (42,949,672.96 seconds - over one year).
Counter	Counters are never writable and cannot be reset. They indicate the count value since the interface was last restarted (<i>sysUpTime</i> = 0). Counters roll over to zero after reaching their maximum value 2^{32} (4,294,967,296).
Integer	A single-precision, signed integer. Valid range is -2^{31} (-2,147,483,648) to $2^{31}-1$ (2,147,483,647).
Integer16	A signed, 16-bit integer. Valid range is -2^{15} (-32,768) to $2^{15}-1$ (32,767).

3.3 – MIB-II Variables

These variables are common to all SNMP implementations. This common set of management variables is defined in the Internet Engineering Task Force (IETF) standards document RFC 1213.

3.3.1 – System

Name	sysDescr
Description	Describes type of interface as ASCII text.
Host Address	0x100000
SNMP Object ID	1.3.6.1.2.1.1.1
Size	75 characters
Type	DisplayString
Attributes	Read-only
Default Value	"Attero Tech CobraNet LE version 2.1.3 LM3S6918"
Implemented Version	2.1.2 - Note: The default value was changed in V2.1.3 to include the software revision level

Name	sysObjectID
Description	The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprise sub-tree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining 'what kind of box' is being managed. For example, if vendor 'Attero Tech' was assigned the sub-tree 1.3.6.1.4.1.2680, the identifier 'CobraNet LE' could be assigned to 1.3.6.1.4.1.2680.1.2.50.1
Host Address	0x100100
SNMP Object ID	1.3.6.1.2.1.1.2
Size	60 characters
Type	OID
Attributes	Read-only
Default Value	1.3.6.1.4.1.2680.1.2.(CobraNet manufacturer ID).(manufacturer product ID)
Implemented Version	2.1.2

Name	sysContact
Description	The identification of the contact person for this managed node, together with information on how to contact this person.
Host Address	0x100200
SNMP Object ID	1.3.6.1.2.1.1.4
Size	60 characters
Type	DisplayString
Attributes	Read/Write - Persistent
Default Value	Zero length string
Implemented Version	2.1.2

Name	sysName
Description	A name assigned to this managed node. By convention, this is the node's fully qualified domain name.
Host Address	0x100300
SNMP Object ID	1.3.6.1.2.1.1.5
Size	60 characters
Type	DisplayString
Attributes	Read/Write - Persistent
Default Value	Product specific
Implemented Version	2.1.2

Name	sysLocation
Description	The physical location of this node (e.g., "telephone closet, 3rd floor")
Host Address	0x100400
SNMP Object ID	1.3.6.1.2.1.1.6
Size	60 characters
Type	DisplayString
Attributes	Read/Write - Persistent
Default Value	Zero length string
Implemented Version	2.1.2

Name	sysUpTime
Description	Time in 100ths of a second since the network management portion of the system was last re-initialized.
Host Address	0x100500
SNMP Object ID	1.3.6.1.2.1.1.3
Type	TimeTicks
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	sysServices
Description	A value which indicates the set of services supported by this entity. The value is a sum. This sum initially takes the value zero. For each layer, L, in the range 1 through 7 that this node performs transactions for, 2 raised to (L - 1) is added. For example, a node which performs primarily routing functions would have a value of 4 which is equal to $(2^{(3-1)})$. A node which is a host offering application services would have a value of 72 or $(2^{(4-1)} + 2^{(7-1)})$. In the context of the Internet suite of protocols, the following service layers are commonly supported: 1 physical (e.g., repeaters), 2 datalink/subnetwork (e.g., bridges), 3 internet (e.g., IP gateways), 4 end-to-end (e.g., IP hosts) and 7 applications (e.g., mail relays). For systems including OSI protocols, layers 5 and 6 may also be counted.
Host Address	0x100502
SNMP Object ID	1.3.6.1.2.1.1.7
Type	Integer
Attributes	Read-only
Default Value	72
Implemented Version	2.1.2

3.3.2 - Interface

Name	ifNumber
Description	The number of network interfaces (regardless of their current state) present on this system.
Host Address	0x110000
SNMP Object ID	1.3.6.1.2.1.2.1
Type	Integer
Attributes	Read-only
Default Value	1
Implemented Version	2.1.2

Name	ifDescr
Description	A string containing information about the interface. This string should include the name of the manufacturer, the product name, and the version of the hardware interface.
Host Address	0x110001
SNMP Object ID	1.3.6.1.2.1.2.2.1.2
Size	Up to 2 characters.
Type	DisplayString
Attributes	Read-only
Default Value	"lm"
Implemented Version	2.1.2

Name	ifType
Description	The type of interface distinguished according to the physical/link protocol(s) immediately 'below' the network layer in the protocol stack. Reference RFC 1213 for all type identifiers
Host Address	0x11000A
SNMP Object ID	1.3.6.1.2.1.2.2.1.3
Type	Integer
Attributes	Read-only
Default Value	6
Implemented Version	2.1.2

Name	ifMtu
Description	The size of the largest datagram or 'packet' that can be sent/received by the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.
Host Address	0x11000B
SNMP Object ID	1.3.6.1.2.1.2.2.1.4
Type	Integer
Attributes	Read-only
Default Value	1500
Implemented Version	2.1.2

Name	ifSpeed
Description	An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.
Host Address	0x11000C
SNMP Object ID	1.3.6.1.2.1.2.2.1.5
Type	Gauge32
Attributes	Read-only
Default Value	100000000
Implemented Version	2.1.2

Name	ifPhysAddress
Description	The interface's address at the protocol layer immediately 'below' the network layer in the protocol stack.
Host Address	0x11000D
SNMP Object ID	1.3.6.1.2.1.2.2.1.6
Type	PhysAddress
Attributes	Read-only
Default Value	N.A.
Implemented Version	2.1.2

Name	ifAdminStatus
Description	The desired state of the interface. The testing(3) state indicates that no operational packets can be passed up(1) - ready to pass packets down(0)
Host Address	0x111000
SNMP Object ID	1.3.6.1.2.1.2.2.1.7
Type	Integer
Attributes	Read/Write
Default Value	1
Implemented Version	2.1.2

Name	ifOperStatus
Description	The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed. up(1) - ready to pass packets down(0)
Host Address	0x112000
SNMP Object ID	1.3.6.1.2.1.2.2.1.8
Type	Integer
Attributes	Read-only
Default Value	1
Implemented Version	2.1.2

Name	ifLastChange
Description	The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.
Host Address	0x112001
SNMP Object ID	1.3.6.1.2.1.2.2.1.9
Type	TimeTicks
Attributes	Read-only
Default Value	N.A.
Implemented Version	2.1.2

Name	ifInOctets
Description	The total number of octets received on the interface, including framing characters.
Host Address	0x112016
SNMP Object ID	1.3.6.1.2.1.2.2.1.10
Type	Counter48
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifInUcastPkts
Description	The number of subnetwork-unicast packets delivered to a higher-layer protocol.
Host Address	0x112018
SNMP Object ID	1.3.6.1.2.1.2.2.1.11
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifInNUcastPkts
Description	The number of non-unicast (i.e., subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.
Host Address	0x112019
SNMP Object ID	1.3.6.1.2.1.2.2.1.12
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifInDiscards
Description	The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their delivery to a higher-layer protocol. One possible reason for discarding such a packet could be lack of buffer space.
Host Address	0x11201A
SNMP Object ID	1.3.6.1.2.1.2.2.1.13
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifInErrors
Description	The number of inbound packets that contained errors preventing delivery to a higher-layer protocol.
Host Address	0x11201B
SNMP Object ID	1.3.6.1.2.1.2.2.1.14
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifInUnknownProtos
Description	The number of packets received which were discarded due to an unknown or unsupported protocol.
Host Address	0x11201C
SNMP Object ID	1.3.6.1.2.1.2.2.1.15
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifOutOctets
Description	The total number of octets transmitted by the interface, including framing characters.
Host Address	0x11201D - 0x11201E
SNMP Object ID	1.3.6.1.2.1.2.2.1.16
Type	Counter48
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifOutUcastPkts
Description	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent.
Host Address	0x11201F
SNMP Object ID	1.3.6.1.2.1.2.2.1.17
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifOutNUcastPkts
Description	The total number of packets that higher-level protocols requested be transmitted to a non-unicast (i.e., a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent.
Host Address	0x112020
SNMP Object ID	1.3.6.1.2.1.2.2.1.18
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifOutDiscards
Description	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their transmission. One possible reason for discarding such a packet could be to free up buffer space.
Host Address	0x112021
SNMP Object ID	1.3.6.1.2.1.2.2.1.19
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifOutErrors
Description	The number of outbound packets that could not be transmitted due to errors.
Host Address	0x112022
SNMP Object ID	1.3.6.1.2.1.2.2.1.20
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ifOutQLen
Description	The length of the output packet queue (in packets).
Host Address	0x112023
SNMP Object ID	1.3.6.1.2.1.2.2.1.21
Type	Integer
Attributes	Read-only
Default Value	N.A.
Implemented Version	2.1.2

Name	ifSpecific
Description	A reference to MIB definitions specific to the particular media being used to implement the interface. For example, if the interface is implemented by Ethernet, then the value of this object refers to a document defining objects specific to Ethernet. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier, and any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.
Host Address	0x112024
SNMP Object ID	1.3.6.1.2.1.2.2.1.22
Size	2
Type	OID
Attributes	Read-only
Default Value	0.0
Implemented Version	2.1.2

3.3.3 - IP

Name	ipForwarding
Description	The indication of whether this entity is acting as an IP gateway to forward datagrams received by, but not addressed to, this entity. IP gateways forward datagrams. IP hosts do not (except those source-routed via the host).
Host Address	0x130000
SNMP Object ID	1.3.6.1.2.1.4.1
Type	Integer
Attributes	Read/Write
Default Value	Always reads 2
Implemented Version	2.1.2

Name	ipDefaultTTL
Description	The default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity when a TTL value is not supplied by the transport layer protocol.
Host Address	0x130001
SNMP Object ID	1.3.6.1.2.1.4.2
Type	Integer
Attributes	Read/Write
Default Value	255
Implemented Version	2.1.2

Name	ipInReceives
Description	The total number of input datagrams received, including those received in error.
Host Address	0x131000
SNMP Object ID	1.3.6.1.2.1.4.3
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipInHdrErrors
Description	The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.
Host Address	0x131001
SNMP Object ID	1.3.6.1.2.1.4.4
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipInAddrErrors
Description	The number of input datagrams discarded because the IP address in the IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP Gateways and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.
Host Address	0x131002
SNMP Object ID	1.3.6.1.2.1.4.5
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipForwDatagrams
Description	The number of input datagrams for which this entity was not the final IP destination, as a result of which an attempt was made to find a route to forward them to the final destination. In entities which do not act as IP Gateways, this counter will include only those packets which were Source-Routed via this entity, and the Source-Route option processing was successful.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.6
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipInUnknownProtos
Description	The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
Host Address	0x131003
SNMP Object ID	1.3.6.1.2.1.4.7
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipInDiscards
Description	The number of IP datagrams received successfully but which were discarded (e.g., for lack of buffer space). Note that this counter does not include datagrams discarded while awaiting re-assembly.
Host Address	0x131004
SNMP Object ID	1.3.6.1.2.1.4.8
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipInDelivers
Description	The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).
Host Address	0x131005
SNMP Object ID	1.3.6.1.2.1.4.9
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipOutRequests
Description	The total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this does not include datagrams counted in ipForwDatagrams.
Host Address	0x131006
SNMP Object ID	1.3.6.1.2.1.4.10
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipOutDiscards
Description	The number of IP datagrams for which no problem existed to prevent their transmission, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.11
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipOutNoRoutes
Description	The number of IP datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes packets counted in ipForwDatagrams which meet this 'no-route' criterion. This includes any datagrams which cannot route because all of its default gateways are down.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.12
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipReasmTimeout
Description	The maximum number of seconds which received fragments are held while they are awaiting reassembly.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.13
Type	Integer
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipReasmReqds
Description	The number of IP fragments received which needed to be reassembled.
Host Address	0x131007
SNMP Object ID	1.3.6.1.2.1.4.14
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipReasmOKs
Description	The number of IP datagrams successfully re-assembled.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.15
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipReasmFails
Description	The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc). Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received. This value will always increment on receipt of a fragmented packet as CobraNet does not support packet re-assembly
Host Address	0x131007 (same as ipReasmReqds)
SNMP Object ID	1.3.6.1.2.1.4.16
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipFragOKs
Description	The number of IP datagrams that have been successfully fragmented at this entity. The CobraNet interface does not support fragmentation. This variable will always read 0.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.17
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipFragFails
Description	The number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be, e.g., because their Don't Fragment flag was set. The CobraNet interface does not support fragmentation. This variable will always read 0.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.18
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipFragCreates
Description	The number of IP datagram fragments that have been generated as a result of fragmentation at this entity. The CobraNet interface does not support fragmentation. This variable will always read 0.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.19
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	ipRoutingDiscards
Description	The number of routing entries which were chosen to be discarded even though they are valid. One possible reason for discarding such an entry could be to free-up buffer space for other routing entries. The CobraNet interface does not support routing. This variable will always read 0.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.4.23
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

3.3.4 – UDP

Name	udpInDatagrams
Description	The total number of UDP datagrams delivered to UDP users.
Host Address	0x140000
SNMP Object ID	1.3.6.1.2.1.7.1
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	udpNoPorts
Description	The total number of received UDP datagrams for which there was no application at the destination port.
Host Address	0x140001
SNMP Object ID	1.3.6.1.2.1.7.2
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	udpInErrors
Description	The number of received UDP datagrams that could not be delivered for reasons other than lack of an application at the destination port.
Host Address	0x140002
SNMP Object ID	1.3.6.1.2.1.7.3
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	udpOutDatagrams
Description	The total number of UDP datagrams sent from this entity.
Host Address	0x140003
SNMP Object ID	1.3.6.1.2.1.7.4
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	udpLocalAddress
Description	The local IP address for this UDP listener. In the case of a UDP listener which is willing to accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used.
Host Address	N.A.
SNMP Object ID	N.A.
Type	IpAddress
Attributes	Read-only
Default Value	N.A.
Implemented Version	Not supported

Name	udpLocalPort
Description	The local port number for this UDP listener.
Host Address	N.A.
SNMP Object ID	N.A.
Type	Integer16
Attributes	Read-only
Default Value	N.A.
Implemented Version	Not supported

3.3.5 – SNMP

Name	snmplnPkts
Description	The total number of SNMP Messages received from the transport service.
Host Address	0x150000
SNMP Object ID	1.3.6.1.2.1.11.1
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmpOutPkts
Description	The total number of SNMP Messages passed to the transport service.
Host Address	0x150001
SNMP Object ID	1.3.6.1.2.1.11.2
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnBadVersions
Description	The total number of SNMP Messages received which were for an unsupported SNMP version.
Host Address	0x150002
SNMP Object ID	1.3.6.1.2.1.11.3
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnBadCommunityNames
Description	The total number of SNMP Messages received which used an unknown community name.
Host Address	0x150003
SNMP Object ID	1.3.6.1.2.1.11.4
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnBadCommunityUses
Description	The total number of SNMP Messages received for which an operation was not allowed by the SNMP community named in the Message.
Host Address	0x150004
SNMP Object ID	1.3.6.1.2.1.11.5
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnASNParseErrs
Description	The total number of ASN.1 or BER errors encountered when decoding received SNMP Messages.
Host Address	0x150005
SNMP Object ID	1.3.6.1.2.1.11.6
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnTooBig
Description	The total number of SNMP PDUs received for which the value of the error-status field was 'tooBig'.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.8
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmplnNoSuchNames
Description	The total number of SNMP PDUs received for which the value of the error-status field was 'noSuchName'.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.9
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmplnBadValues
Description	The total number of SNMP PDUs received for which the value of the error-status field was 'badValue'.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.10
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmplnReadOnlys
Description	The total number of valid SNMP PDUs received for which the value of the error-status field was 'readOnly'. Note that it is a protocol error to generate an SNMP PDU which contains the value 'readOnly' in the error-status field. This object is provided as a means of detecting incorrect implementations of SNMP.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.11
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmplnGenErrs
Description	The total number of SNMP PDUs received for which the value of the error-status field was 'genErr'.
Host Address	Not Available
SNMP Object ID	1.3.6.1.2.1.11.12
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmplnTotalReqVars
Description	The total number of MIB objects which have been retrieved successfully as a result of receiving valid SNMP Get-Request and Get-Next PDUs.
Host Address	0x150006
SNMP Object ID	1.3.6.1.2.1.11.13
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnTotalSetVars
Description	The total number of MIB objects which have been altered successfully as a result of receiving valid SNMP Set-Request PDUs.
Host Address	0x150007
SNMP Object ID	1.3.6.1.2.1.11.14
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnGetRequests
Description	The total number of SNMP Get-Request PDUs which have been accepted and processed.
Host Address	0x150008
SNMP Object ID	1.3.6.1.2.1.11.15
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnGetNexts
Description	The total number of SNMP Get-Next PDUs which have been accepted and processed.
Host Address	0x150009
SNMP Object ID	1.3.6.1.2.1.11.16
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnSetRequests
Description	The total number of SNMP Set-Request PDUs which have been accepted and processed.
Host Address	0x15000A
SNMP Object ID	1.3.6.1.2.1.11.17
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmplnGetResponses
Description	The total number of SNMP Get-Response PDUs which have been accepted and processed.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.18
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmplnTraps
Description	The total number of SNMP Trap PDUs which have been accepted and processed.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.19
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmpOutTooBigs
Description	The total number of SNMP PDUs generated for which the value of the error-status field was 'tooBig'.
Host Address	0x15000B
SNMP Object ID	1.3.6.1.2.1.11.20
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmpOutNoSuchNames
Description	The total number of SNMP PDUs generated for which the value of the error-status was 'noSuchName'.
Host Address	0x15000C
SNMP Object ID	1.3.6.1.2.1.11.21
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmpOutBadValues
Description	The total number of SNMP PDUs generated for which the value of the error-status field was 'badValue'.
Host Address	0x15000D
SNMP Object ID	1.3.6.1.2.1.11.22
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmpOutGenErrs
Description	The total number of SNMP PDUs generated for which the value of the error-status field was 'genErr'.
Host Address	0x15000E
SNMP Object ID	1.3.6.1.2.1.11.24
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmpOutGetRequests
Description	The total number of SNMP Get-Request PDUs generated.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.25
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmpOutGetNexts
Description	The total number of SNMP Get-Next PDUs generated.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.26
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmpOutSetRequests
Description	The total number of SNMP Set-Request PDUs generated.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.27
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmpOutGetResponses
Description	The total number of SNMP Get-Response PDUs generated.
Host Address	0x150001 (same as snmpOutPackets)
SNMP Object ID	1.3.6.1.2.1.11.28
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	snmpOutTraps
Description	The total number of SNMP Trap PDUs generated.
Host Address	Not available
SNMP Object ID	1.3.6.1.2.1.11.29
Type	Counter
Attributes	Read-only
Default Value	Always reads 0
Implemented Version	2.1.2

Name	snmpEnableAuthenTraps
Description	Indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The value of this object overrides any configuration information. It provides a means to disable all authentication-failure traps.
Host Address	0x15000F
SNMP Object ID	1.3.6.1.2.1.11.30
Type	Integer
Attributes	Read-only
Default Value	Always reads 2
Implemented Version	2.1.2

3.4 – CobraNet Variables

3.4.1 – Firmware

Name	firmwareProtocolVersion
Description	Highest CobraNet protocol version supported by firmware. Current protocol version is 2. A protocol version of 0 indicates an unsupported test version of firmware.
Host Address	0x0
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.1
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	firmwareMajorVersion
Description	CobraNet firmware major revision number.
Host Address	0x1
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.2
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	firmwareMinorVersion
Description	CobraNet firmware minor revision number.
Host Address	0x2
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.3
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	firmwareBootVersion
Description	Configuration record revision number. The configuration record contains the MAC address for the interface and other permanent initialization parameters.
Host Address	0x3
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.4
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	firmwareMfgId
Description	Identifies the manufacturer of the CobraNet device. 0 represents unknown. Attero Tech is 50.
Host Address	0x4
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.5
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	firmwareMfgProductId
Description	Identifies product type per manufacturer. Product identifiers are unique per manufacturer identifier. 0 represents unknown product. CobraNet LE is 1.
Host Address	0x5
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.6
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	firmwareMfgVersion
Description	A manufacturer assigned minor revision number for firmware. A non-zero value indicates manufacturer has made some modification to the standard firmware as released by Cirrus Logic.
Host Address	0x6
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.7
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	firmwareRestart
Description	<p>Reboots the interface when set to a non-zero value. Invoking this function will cause loss of communications. The interface will attempt to send a response to an SNMP set request before restarting. Due to the nature of SNMP, receipt of such a response by the manager is not guaranteed. Invoking this feature via SHMI will cause the SHMI to become inoperable until the interface has completed re-initialization. A successful restart can be verified by reading the <i>sysUpTime</i> variable. <i>sysUpTime</i> returns to 0 following a restart.</p> <p>A restart via SNMP may adversely affect an SHMI connected host processor. Care should be taken to ensure that a host processor attempting to communicate via SHMI during reset can recover from the SHMI's failure to respond and, further, that the host processor will continue to function properly following re-initialization of the MI variables to their default or persistent values.</p>
Host Address	0x100
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.8
Type	Integer
Attributes	Read/Write
Default Value	0
Implemented Version	2.1.2

Name	firmwareFreeCycles
Description	Free CPU cycles in thousandths. For example, 475 means 47.5% of CPU cycles are free, i.e., running idle loop.
Host Address	0x9
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.11
Type	Integer
Attributes	Read-only
Implemented Version	Currently disabled for CobraNet LE

3.4.2 – Hardware Identification

Different platforms and hardware versions have different channel and processing capacities. By recognizing the capabilities of the interface, the host may optimize the configuration to take advantage of the capabilities.

Before the introduction of the variables documented below, the host could use sysDescr, sysObjectID, and firmwareVersion to determine the capabilities. With the introduction of the CM-1 rev F, these variables no longer fully characterize the interface. CM-1 rev F features a memory speed increase and thus a capacity improvement over CM-1 rev E and previous revisions.

Name	firmwareHardwarePlatform
Description	<p>CobraNet interface hardware platform.</p> <ul style="list-style-type: none"> 1 - RAVE 2 - Reference design 3 - High capacity reference design 4 - CM-1 with AMD flash memory (alternate supplier) 5 - CM-1 with Micron flash memory (standard supplier) <ul style="list-style-type: none"> 18100 - CS18100 18101 - CS18101 18102 - CS18102 18110 - CS18110 18111 - CS18111 18112 - CS18112 6965 - LM3S6965 6911 - LM3S6911 6918 - LM3S6918 6938 - LM3S6938
Host Address	0x7
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.9
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	firmwareHardwareVersion
Description	<p>Version number specific to platform.</p> <ul style="list-style-type: none"> 1 - Reported for all RAVE and reference design hardware as these platforms lack hardware version identification feature: CM-1 rev A-C (95Mhz) and CS18101 rev 1 (prototypes). 2 - CM-1 rev C-E (100 MHz-capable), CS18xxx rev 2. 3 - CM-1 rev F (1 WS-capable) 4 - CobraNet LE
Host Address	0x8
SNMP Object ID	1.3.6.1.4.1.2680.1.1.1.10
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

3.4.3 - Flash

Name	flashTotalSize
Description	Total flash memory size in bytes.
Host Address	0x1000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.2.1
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	flashSectorSize
Description	Largest flash sector size in bytes.
Host Address	0x1001
SNMP Object ID	1.3.6.1.4.1.2680.1.1.2.2
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	flashPersistSequence
Description	Sequence number for persistence storage. This gives an approximate indication of the wear on the flash memory from persistent stores. The sequence number is incremented each time a sector is erased to make room for a persistent store. Typically two sectors are used for persistent storage. The sequence number divided by the number of sectors used for persistence yields the approximate erase cycle count each sector has experienced. Flash memory is typically rated for no less than 100,000 erase cycles.
Host Address	0x1002
SNMP Object ID	1.3.6.1.4.1.2680.1.1.2.3
Type	Counter
Attributes	Read-only
Implemented Version	2.1.2

Name	flashPersistType
Description	Indicates the type identifier for the persistent store dataset. If firmware is updated to a version that uses a different dataset type (or different size), persistent settings will be lost.
Host Address	0x1003
SNMP Object ID	1.3.6.1.4.1.2680.1.1.2.4
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	flashPersistStores
Description	The number of times variables have been written to flash during sysUpTime. Use of flashPersistAck is preferred to determine completion of a persistent save operation.
Host Address	0x1005
SNMP Object ID	1.3.6.1.4.1.2680.1.1.2.6
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	flashPersistEnable
Description	Non-zero value enables variable persistence feature. Read/Write - Persistent type variables will be automatically written to non-volatile memory when changed. Values will be restored on power-up.
Host Address	0x1100
SNMP Object ID	1.3.6.1.4.1.2680.1.1.2.7
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

Name	flashPersistAck
Description	Forces a write of variables to non-volatile memory when set to a non-zero value. Value returns to zero when write has completed. This value will not change if persistence is disabled. This feature is recommended for use during factory configuration where writing and confirmation of success must be done in a timely manner. In normal use it is best to let the interface schedule writes to non-volatile memory. Over-use of this feature can result in excessive wear on the flash device.
Host Address	0x1200
SNMP Object ID	1.3.6.1.4.1.2680.1.1.2.8
Type	Integer
Attributes	Read/Write
Default Value	0
Implemented Version	2.1.2

3.4.4 – Errors

Name	errorCode
Description	Last 4 error codes reported. This is implemented as a 4 byte FIFO with the most recent error in the lower byte. If the same error occurs multiple times in a row it will only be put into the FIFO once. See error table.
Host Address	0x2002
SNMP Object ID	1.3.6.1.4.1.2680.1.1.3.3
Type	Integer
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	errorCount
Description	Number of errors reported during sysUpTime.
Host Address	0x2003
SNMP Object ID	1.3.6.1.4.1.2680.1.1.3.4
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	modeRateStatus
Description	Indicates the latency and sample rate operating mode currently in effect for the interface.
Host Address	0x2005
SNMP Object ID	1.3.6.1.4.1.2680.1.1.3.6.2
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	modeRateControl
Description	Selects latency and sample rate mode for the interface. The following modes are supported: 0x600 - 5-1/3 ms latency, 48 kHz sample rate. This was a Read/Write - Persistent variable, but CobraNet LE only supports one value so it was changed to read-only
Host Address	0x2100
SNMP Object ID	1.3.6.1.4.1.2680.1.1.3.6.1
Type	Integer
Attributes	Read-only
Default Value	0x600
Implemented Version	2.1.2

3.4.5 – Conductor

These variables determine the values transmitted in the header of the Beat Packet when the CobraNet interface is acting as the network conductor.

Name	conductorCycleRate
Description	Number of isochronous cycles per second as a 16.16 fixed point number. This is a legacy variable. It always reports default value and should not be changed.
Host Address	0x10000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.4.1
Type	Integer48
Attributes	Read-only
Default Value	750
Implemented Version	2.1.2

Name	conductorPriority
Description	Specifies the conductor priority for a CobraNet device. The device with the highest priority will become the conductor for the network. CobraNet LE cannot be a conductor so it will always be 0 0 - Never Conduct
Host Address	0x10002
SNMP Object ID	1.3.6.1.4.1.2680.1.1.4.2
Type	Integer16
Attributes	Read-only
Default Value	0x00
Implemented Version	2.1.2

Name	conductorGaps
Description	These timing specifications are applicable only to repeater networks and therefore do not apply to most CobraNet applications. This variable is changed on the active conductor in order to reduce the probability of collisions. In most cases the default value works well. Useful values other than the default are dependant on the specific network topology used. Channel Gap in LS byte - Larger channel gaps allow greater network diameter. Packet Gap in MS byte - Larger packet gaps increase resilience to unregulated traffic. Changed from Read/Write - Persistent to read-only because CobraNet LE cannot be a conductor.
Host Address	0x10003
SNMP Object ID	1.3.6.1.4.1.2680.1.1.4.3
Type	Integer16
Attributes	Read-only
Default Value	0x0306
Implemented Version	2.1.2

Name	conductorStatus
Description	Conductor status: 0 - This interface is not the conductor 1 - This interface is the conductor
Host Address	0x11000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.4.4
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

3.4.6 – Conductor Information

These conductor variables give a description and status of the current conductor of the CobraNet network.

Name	condInfoPriority
Description	Conductor priority of the current conductor. 1 - Lowest conductor priority 255 - Highest conductor priority If the priority of the current conductor is adjusted, condInfoPriority will change to reflect this. Dependent on priority setting of other devices on the network, a change in current conductor priority does not necessarily induce a change in conductor and condInfoMAC.
Host Address	0x11001
SNMP Object ID	1.3.6.1.4.1.2680.1.1.4.5
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	condInfoMAC
Description	Ethernet MAC address of the current conductor. condInfoMAC should read 00:00:00:00:00:00 if there is no conductor on the network. There is no conductor if there is 0 or 1 CobraNet device(s) attached to the network or if all CobraNet devices have a condPriority setting of 0.
Host Address	0x11002
SNMP Object ID	1.3.6.1.4.1.2680.1.1.4.6
Type	Physical Address
Attributes	Read-only
Implemented Version	2.1.2

Name	condInfoLastChange
Description	sysUpTime value at time of last change to condInfoMAC.
Host Address	0x11005
SNMP Object ID	1.3.6.1.4.1.2680.1.1.4.7
Type	Time Ticks
Attributes	Read-only
Implemented Version	2.1.2

Name	condInfoChanges
Description	Count of condInfoMAC changes since boot. A single conductor arbitration event may produce multiple increments.
Host Address	0x11007
SNMP Object ID	1.3.6.1.4.1.2680.1.1.4.8
Type	Counter
Attributes	Read-only
Implemented Version	2.1.2

3.4.7 – Packet Bridge

Not Implemented in CobraNet LE

3.4.8 – Serial Bridge

Name	serialFormat
Description	This variable is used to enable or disable the Serial Communications Interface. The data format for both transmit and receive directions is 8 data bits, no parity, and 1 stop bit. 0x01 - Enable serial bridging.
Host Address	0x24000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.10.1.1
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

Name	serialBaud
Description	Baud rate for transmission and reception. The baud rate is specified in bits per second. The minimum baud rate is 600 baud. Maximum baud rate is 57,600.
Host Address	0x24001
SNMP Object ID	1.3.6.1.4.1.2680.1.1.10.1.2
Type	Integer
Attributes	Read/Write - Persistent
Default Value	19200
Implemented Version	2.1.2

Name	serialPPeriod
Description	<p>Time in 256ths of a millisecond before a character received at the SCI port is placed in a packet and transmitted. Shorter periods to achieve lower latency are appropriate for real time connections such as MIDI. With higher settings more characters can be packed into a packet before it is transmitted resulting in increased efficiency. Higher settings are recommended for bulk data transfer applications.</p> <p>The isochronous cycle period (1-1/3 mS) determines the minimum serialPPeriod. Setting serialPPeriod below the isochronous cycle rate does not further improve responsiveness. The upper limit of responsiveness can also be affected by control channel accessibility and pipeline delays. The depth of the receive SCI character queue in combination with the baud rate determines the maximum allowed setting. The character buffer can accommodate 100 characters. This allows for operation at the default 10ms period at a baud rate of 57,600. At this baud rate, larger settings will result in buffer overflow and loss of data.</p>
Host Address	0x24002
SNMP Object ID	1.3.6.1.4.1.2680.1.1.10.1.3
Type	Integer
Attributes	Read/Write - Persistent
Default Value	2560 (10ms)
Implemented Version	2.1.2

Name	serialRxMAC
Description	<p>MAC address of the CobraNet Interface from which SCI data will be accepted. This may be any multicast address though 01:60:2B:FD:00:00 through 01:60:2B:FD:FF:FF have been reserved by Cirrus Logic for use as "asynchronous global channels." ifPhysAddress is the only usable unicast address (CobraNet does not support Ethernet promiscuous mode).</p>
Host Address	0x24003
SNMP Object ID	1.3.6.1.4.1.2680.1.1.10.1.4
Type	PhysAddress
Attributes	Read/Write - Persistent
Default Value	01:60:2B:FD:00:00
Implemented Version	2.1.2

Name	serialTxMAC
Description	<p>MAC address of the CobraNet interface to which serial data is sent. May be any multicast or unicast address.</p>
Host Address	0x24100
SNMP Object ID	1.3.6.1.4.1.2680.1.1.10.1.7
Type	PhysAddress
Attributes	Read/Write - Persistent
Default Value	01:60:2B:FD:00:00
Implemented Version	2.1.2

3.4.9 – Interrupt Control

Not currently implemented in CobraNet LE.

3.4.10 – Audio

Not currently implemented in CobraNet LE

3.4.11 – Receivers

Name	rxStatus
Description	Indicates bundle reception. Bundle reception does not necessarily indicate audio reception. Consult rxSubFormat variables for audio reception status. 0 - Bundle is not being received 1 - Bundle is being received
Host Address	0x4n000 (n is 0 based receiver number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.6.1.1.6.n (n is 1 based receiver number)
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	rxDropouts
Description	Counts number of times bundle reception has been interrupted. Interruptions can be caused by transmitter failure or by reconfiguring the receiver. This variable is implemented by counting transitions to 0 of rxStatus.
Host Address	0x4n001 (n is 0 based receiver number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.6.1.1.7.n (n is 1 based receiver number)
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	rxDelay
Description	<p>Indicates additional group delay imposed on the received audio due to network forwarding delays. Delay is expressed in units of audio transmission cycles (1-1/3ms for standard 5-1/3ms latency mode, 2/3ms for 2-2/3ms latency mode, and 1/3ms for 1-1/3ms latency mode).</p> <p>Forwarding delay is continuously monitored by the receiver. If forwarding delay changes due to a network reconfiguration or change in <i>rxBundle</i>, the receiver delay will adapt to the new conditions. A discontinuity in the audio stream will be experienced whenever the receiver delay is adjusted in this manner.</p> <p>Normal propagation delay is 4 isochronous cycle periods. This normal condition is indicated by a 0 reading in <i>rxDelay</i>. A reading of 1 indicates an additional isochronous cycle period delay (for a total of 5 cycles) has been inserted due to network forwarding delay.</p>
Host Address	0x4n002 (n is 0 based receiver number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.6.1.1.8.n (n is 1 based receiver number)
Type	Integer
Attributes	Read-only
Implemented Version	Not currently implemented in CobraNet LE

Name	rxMinDelay
Description	<p>Selects a minimum additional delay imposed on the received audio. Delay is expressed in units of isochronous cycles (1-1/3ms for standard 5-1/3ms latency mode, 2/3ms for 2-2/3ms latency mode and 1/3ms for 1-1/3ms latency mode). This variable is designed to allow configuration of a deterministic common delay for all CobraNet interfaces in larger network installations. <i>rxDelay</i> will never be reduced below this setting. This variable is not designed for actively delaying audio for architectural applications. The maximum setting for <i>rxMinDelay</i> is determined by the amount of Ethernet packet buffering available on the interface. Excessive settings will result in <code>ERROR_RXBUFFER_OVERFLOW</code> errors and accompanying audio dropouts.</p>
Host Address	0x4n106 (n is 0 based receiver number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.6.1.1.9.n (n is 1 based receiver number)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	Not currently implemented in CobraNet LE

Name	rxSubFormat
Description	Vector of received audio format for each sub-channel. See Table 1 for a complete listing of valid format values. The least significant bit of these variables is set when the received format is supported for reception by the CobraNet interface. A test of this least significant bit can be used to determine correct reception on a per audio channel basis. All entries in this vector will be 0 if rxStatus is zero.
Host Address	0x4n30m (n is 0 based receiver number, m is 0 based audio channel number)
Count	8
SNMP Object ID	1.3.6.1.4.1.2680.1.1.6.2.1.3.n.m (n is 1 based receiver number, m is the 1 based sub-channel number)
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	rxBundle
Description	Receive bundle assignment.
Host Address	0x4n100 (n is 0 based receiver number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.6.1.1.2.n (n is 1 based receiver number)
Type	Integer16
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

Name	rxSubMap
Description	Audio routing channel destinations for each audio channel in a received bundle.
Host Address	0x4n20m (n is 0 based receiver number, m is 0 based audio channel number)
Count	8
SNMP Object ID	1.3.6.1.4.1.2680.1.1.6.2.1.2.n.m (n is 1 based receiver number, m is the 1 based audio channel number)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	First receiver {33, 34, 35, 36, 37, 38, 39, 40} Second receiver {41, 42, 43, 44, 45, 46, 47, 48}
Implemented Version	2.1.2

3.4.12 - Transmitters

Name	txDropouts
Description	Count of times channel transmission has been interrupted. Interruptions can be caused by loss of transmit permission from conductor or by changes to txBundle. Implemented by counting transitions to 0 of txPosition.
Host Address	0x5n000 (n is 0 based transmitter number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.7.1.1.7.n (n is 1 based transmitter number)
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	txReceivers
Description	Number of receivers requesting this bundle. This may not be valid for multicast bundles as receivers of multicast bundles are not required to issue a reverse reservation although all receivers currently do. A transmitter will track no more than 4 receivers. txReceivers will never exceed 4.
Host Address	0x5n002 (n is 0 based transmitter number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.7.1.1.9.n (n is 1 based transmitter number)
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	txBundle
Description	Transmit bundle assignment.
Host Address	0x5n100 (n is 0 based transmitter number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.7.1.1.2.n (n is 1 based transmitter number)
Type	Integer16
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

Name	txSubCount
Description	Number of audio channels to transmit in a bundle. Valid values are 0 through 8. Reducing txSubCount is the preferred means for transmitting bundles with less than the maximum 8 audio channels. Short bundles may also be transmitted by setting txSubFormat or txSubMap entries to 0.
Host Address	0x5n105 (n is 0 based transmitter number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.7.1.1.5.n (n is 1 based transmitter number)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	8
Implemented Version	2.1.2

Name	txUnicastMode
Description	Specifies the number of unicast destinations served before automatically switching to multicast bundle transmission. Multicast transmission is useful for efficient point to multipoint routing. However, multicast addressing consumes bandwidth on all ports on a switched network. This variable allows control of multicast traffic from transmitters. 0 - Multicast addressing used at all times. Note: multicast bundles do not transmit data until a receiver is assigned to the same bundle number. 1 - Unicast addressing used to single receiver. Multicast addressing used for multiple receivers. 0x7FFFFF - Multicast addressing is never used. Maximum number of unicast destinations is set by txMaxUnicast. Receiver request priority is used to determine which receivers are serviced if multiple receivers are assigned to this bundle.
Host Address	0x5n107 (n is 0 based transmitter number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.7.1.1.10.n (n is 1 based transmitter number)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0x7FFFFF
Implemented Version	2.1.2

Name	txMaxUnicast
Description	<p>Specifies maximum number of unicast destinations supported simultaneously by the transmitter. Receivers in excess of this setting will not receive the bundle.</p> <p>A transmitter can service up to 4 receivers. The number of unicast destinations transmitted to will never exceed this internal capacity limitation.</p> <p>If <i>txUnicastMode</i> is set lower than <i>txMaxUnicast</i>, the bundle will switch to multicast before the limitation on unicast destinations is reached.</p> <p>If <i>txUnicastMode</i> is set equal to <i>txMaxUnicast</i>, the bundle will switch to multicast when the limitation on unicast destinations is exceeded.</p>
Host Address	0x5n108 (n is 0 based transmitter number)
SNMP Object ID	1.3.6.1.4.1.2680.1.1.7.1.11.n (n is 1 based transmitter number)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	1
Implemented Version	2.1.2

Name	txSubMap
Description	Transmit audio channel (channel within bundle) to audio routing channel (channel of SSI) mapping. This vector contains the routing channel source specifiers per audio channel in the transmitted bundle.
Host Address	0x5n20m (n is 0 based transmitter number, m is 0 based sub-channel number)
Count	8
SNMP Object ID	1.3.6.1.4.1.2680.1.1.7.2.1.2.n .m (n is 1 based transmitter number, m is 1 based audio channel)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	First transmitter {1, 2, 3, 4, 5, 6, 7, 8} Second transmitter {1, 2, 3, 4, 5, 6, 7, 8}
Implemented Version	2.1.2

Name	txSubFormat
Description	Specifies data format for each sub-channel in the transmitted bundle. Please see Table 1 for a complete list of valid format values. <i>modeRateControl</i> must also be set correctly to support the configured format.
Host Address	0x5n30m - 0x5n30m (n is 0 based transmitter number, m is 0 based sub-channel number)
Count	8
SNMP Object ID	1.3.6.1.4.1.2680.1.1.7.2.1.3.n .m (n is 1 based transmitter number, m is 1 based sub-channel number)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0x54000
Implemented Version	2.1.2

3.4.13 – Synchronization

Name	syncConductorClock
Description	Selects sample clock source when acting as network conductor. 0x0 - Internal mode. 0x1 - External Word Clock mode. 0x10 - Not supported on CobraNetSilicon Series devices (Internal with External Sample Synchronization mode). 0x14 - Not supported on CobraNet Silicon Series devices (External Master Clock with External Sample Synchronization mode)
Host Address	0x60000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.8.1
Type	Integer
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	syncPerformerClock
Description	Selects sample clock source when acting as a performer. Values are the same as documented above for syncConductorClock. External clocks applied must be externally synchronized to the conductor. This was changed from Read/Write - Persistent because the CobraNet LE always Phase locks to the network conductor clock.
Host Address	0x60001
SNMP Object ID	1.3.6.1.4.1.2680.1.1.8.2
Type	Integer
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	syncStatus
Description	Indicates current audio clock synchronization status. The following values may be ORed together: Unspecified bits should be ignored. 0x01 - Locked to external or network clock reference. 0x02 - Valid clock present at REFCLK_IN. 0x04 - MUTE is not asserted. Indicates proper operation of CobraNet interface as MUTE is asserted on detection of a fault condition or loss of connection to the network.
Host Address	0x61000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.8.5
Type	Integer
Attributes	Read-only
Default Value	N.A.
Implemented Version	2.1.2

Name	syncCounter
Description	Incremented each time network sync is lost.
Host Address	0x61001
SNMP Object ID	1.3.6.1.4.1.2680.1.1.8.6
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	syncNTime
Description	CobraNet network time. Advances 256 every 1-1/3ms. Network time rolls over after reaching 0xFFFF00 (16,776,960) on 24- and 32-bit platforms.
Host Address	0x61002
SNMP Object ID	1.3.6.1.4.1.2680.1.1.8.7
Type	Integer (Only lower 24 bits are valid. Upper byte is always zero.)
Attributes	Read-only
Implemented Version	2.1.2

3.4.14 – SNMP Monitor

Not currently implemented in CobraNet LE

3.4.15 – MI Monitor

Name	miMonDirty
Description	Incremented when a management interface variable is modified either via SNMP or SHMI. Multiple modifications may result in a single increment of this counter. The act of writing a variable, even if written with its current value, is considered a modification for the purposes of this counter.
Host Address	0x71000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.1.1
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	miMonSNMPDirty
Description	Incremented if a management interface variable is modified through SNMP. The counter is intended to allow detection of variable modification by any SNMP manager. Multiple modifications may result in a single increment of this variable. The act of setting a variable, even if set to its current value, is considered a modification for the purposes of this counter.
Host Address	0x71001
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.1.2
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

Name	miMonHMIDirty
Description	Incremented if a management interface variable is modified through SHMI. The counter is intended to allow detection of variable modification by a local manager. Multiple modifications may result in a single increment of this variable. The act of writing a variable, even if written with its current value, is considered a modification for the purposes of this counter.
Host Address	0x71002
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.1.3
Type	Counter
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

3.4.16 – IP Monitor

Name	ipMonCurrentIP
Description	<p>The current IP address for the CobraNet interface. Changing the current IP address has an immediate effect on IP communications. A value of 0.0.0.0 indicates no IP address assignment for the interface. An IP address can be assigned (or reassigned) to the interface by any of the following means:</p> <p>A value loaded from <i>ipMonStaticIP</i> during power-up.</p> <p>A host processor writing to <i>pMonCurrentIP</i> via the SHMI.</p> <p>Receipt of a BOOTP response packet (typically in response to a transmitted BOOTP request)</p> <p>Receipt of a RARP response packet (RARP requests are not transmitted)</p>
Host Address	0x72000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.2.2
Type	IpAddress
Attributes	Read/Write
Default Value	ipMonStaticIP
Implemented Version	2.1.2

Name	ipMonStaticIP
Description	A power-up static IP address assignment for the interface. A value of 0.0.0.0 indicates no power-up IP address assignment.
Host Address	0x72002
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.2.1
Type	IpAddress
Attributes	Read/Write - Persistent
Default Value	0.0.0.0
Implemented Version	2.1.2

3.4.17 – IF Monitor

MI interface for monitoring redundant Ethernet connection (Dual Link) feature. These variables are only available on the CM-1, CM-2, and CS4961xx/CS1810xx-based hardware.

Name	ifmCurrentIf
Description	Index of the current, active Ethernet connection. 1 - Primary 2 - Secondary On platforms with only one interface, this value will always be one. CobraNet LE always reports one.
Host Address	0x73000
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.3.1
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	ifmLastChange
Description	The value of sysUpTime at the time the ifmCurrentIf was established. If the current state was entered prior to or concurrent with the last re-initialization of the local network management subsystem, then this value will be zero. On platforms with only one interface, this value will always be zero.
Host Address	0x73001
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.3.2
Type	Time Ticks
Attributes	Read-only
Implemented Version	2.1.2

Name	ifmSwitchMode
Description	Controls DualLink behavior. This variable applies only to interfaces which support two Ethernet ports. 0 - (default) Automatic switchover to secondary on failure of primary. 1 - Always use primary port. 2 - Always use secondary port. 3 - If one port is working properly and the other has failed, switch to the port that is working properly. If both are working properly or both have failed, do not change ports. On boot, start with the primary port. On platforms with only one interface, a value of "3" is ignored.
Host Address	0x73100
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.3.4
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	ifmtStatus
Description	Status of Ethernet connection. The following values may be OR'd together: 1 - Ethernet link established 2 - Connection is full-duplex 4 - Ethernet packets being received at a rate of at least 1 packet every two seconds.
Host Address	0x74n00 where: n = 0 = primary Ethernet interface n = 1 = secondary Ethernet interface
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.3.3.1.2.n
Count	2
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

Name	ifmtLastChange
Description	The value of sysUpTime at the time ifmtStatus was established. If the current state was entered prior to or concurrent with the last re-initialization of the local network management subsystem, then this object contains a zero value.
Host Address	0x74n01 where: n = 0 = primary Ethernet interface n = 1 = secondary Ethernet interface
SNMP Object ID	1.3.6.1.4.1.2680.1.1.9.3.3.1.3.n
Count	2
Type	Time Ticks
Attributes	Read-only
Implemented Version	2.1.2

3.5 – DSP Extensions

Not implemented in CobraNet LE.

3.6 – User Extensions

Name	stdUserVersionMajor
Description	Application specific version number; major number. Intended to allow users to identify the contents of the user variable area.
Host Address	0x7e000
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.1.1
Type	Integer
Attributes	SNMP: Read-only SHMI: Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

Name	stdUserVersionMinor
Description	Application specific version number; minor number. Intended to allow users to identify the contents of the user variable area.
Host Address	0x7e001
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.1.2
Type	Integer
Attributes	SNMP: Read-only SHMI: Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

Name	stdUserString
Description	Application specific strings.
Host Address	0x7e100+(n*0x16) where n is 0-based string number
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.2.1.2.n (n is 1-based string number)
Size	60 characters
Count	4
Type	DisplayString
Attributes	Read/Write - first 2 strings are Persistent
Default Value	Zero length string
Implemented Version	2.1.2

Name	stdUserInteger
Description	Application specific integers.
Host Address	0x7e300+n (n is 0-based integer number)
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.1.2.n (n is 1-based integer number)
Count	64
Type	Integer
Attributes	Read/Write - first 32 are Persistent
Default Value	0
Implemented Version	2.1.2

Name	stdUserGPIODirection
Description	User GPIO pins direction control. The lower 8 bits control the direction of the 8 GPIO pins. A 0 bit makes the pin an input, a 1 bit makes the pin an output.
Host Address	0x7e400
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.4.1
Type	Integer
Attributes	Read/Write
Default Value	0
Implemented Version	2.1.2

Name	stdUserGPIOData
Description	User GPIO pins. The lower 8 bits when written will output the bits if the direction is set to 1. The lower 8 bits will read all 8 GPIO pins.
Host Address	0x7e404
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.4.2
Type	Integer
Attributes	Read/Write
Default Value	0
Implemented Version	2.1.2

Name	stdUserADCCtrl
Description	User ADC control. Each of the ADC readings is averaged over 64 samples to reduce noise issues. The samples are taken every 2 beat packets or every 2.666ms on average. The control variable has the following meaning 0 = off 1 = Perform readings but do not send UDP packet with readings > 1 Perform readings and send UDP packet with readings every control count of beat packets. For example, a 2 would cause a UDP packet to be sent every other Beat cycle.
Host Address	0x7e500
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.5.1
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

Name	stdUserADCMAC
Description	User ADC MAC address of where to send the UDP ADC reading packets.
Host Address	0x7e504
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.5.2
Type	PhysAddress
Attributes	Read/Write - Persistent
Default Value	01:60:2B:FD:00:00
Implemented Version	2.1.2

Name	stdUserADCPort
Description	User ADC UDP port number. The ADC readings can also be sent in UDP packets. This is the UDP port number to use. Private UDP port numbers start at 49152.
Host Address	0x7e50A
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.5.3
Type	Integer
Attributes	Read/Write - Persistent
Default Value	65000
Implemented Version	2.1.2

Name	stdUserADCReading
Description	<p>User ADC readings. There can be 9 readings read back for DAC0 thru DAC7 and the internal processor temperature. The valid values are 10 bits. If the value returned is 0xFFFF then it is not a valid value.</p> <p>UDP packet user data:</p> <ul style="list-style-type: none"> 16 bits DAC0 16 bits DAC1 16 bits DAC2 16 bits DAC3 16 bits DAC4 16 bits DAC5 16 bits DAC6 16 bits DAC7 16 bits Processor internal temp <p>Processor temp:</p> $T = ((2.7V - (ADC\ reading * (3/1024))) * 75) - 55$ <p>ex:</p> $T = ((2.7 - (421 * (3/1024))) * 75) - 55 = 55$
Host Address	0x7e600+n (n is 0-based integer number) (0-8)
SNMP Object ID	1.3.6.1.4.1.2680.1.3.3.5.4.n (n is 1-based reading number) (1-9)
Type	Integer
Attributes	Read/Write
Default Value	0xFFFF
Implemented Version	2.1.2

3.7 – System Extensions

Name	sysMAC
Description	The system MAC variable was added to allow setting the MAC address of the CobraNet LE board. It should be set once during production. The address is saved at the last 6 bytes of Persistent memory
Host Address	0x7f000
SNMP Object ID	Not available via SNMP
Type	PhysAddress
Attributes	Read/Write - Persistent
Default Value	FF:FF:FF:FF:FF:FF
Implemented Version	2.1.2

4 – SHMI message protocol

4.1 – Terminology

The following table describes abbreviations, acronyms, and technical terms used.

Term	Meaning
Ack	Positive Acknowledgement
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
HMI	Host Management Interface
SHMI	Serial Host Management Interface
IP	Internet Protocol
bps	Bits Per Second
MAC	Media Access Controller
MI	Management Interface
Nack	Negative Acknowledgement
SCI	Serial Communications Interface
SSI	Synchronous Serial Interface

4.2 – System Description

The CobraNet LE SHMI provides a friendly user interface allowing the user to request information about internal variables and set these variables via a serial port. The SHMI enables the user to easily add CobraNet capability to a given system by allowing the system to request CobraNet information and set CobraNet parameters without necessarily needing to know the details of the variable addresses. The SHMI receives messages via the serial port. The SHMI translates these serial commands to the appropriate CobraNet commands and sets or reads the CobraNet Management variables, responding with an *Ack/Nack* via the serial port.

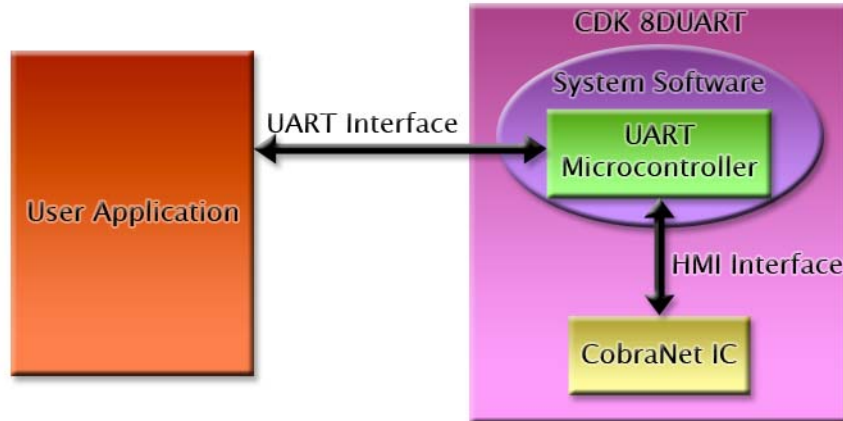


Figure 1 - System Block Diagram

4.3 – Serial Format

The SHMI interface configuration uses 8 data bits, no parity bit, 1 stop bit, and no flow control at a default baud rate of 57,600 bits per second (bps). Of those parameters, only the baud rate can be changed. Acceptable values for the baud rate are 9600, 19200, 38400, and 57600.

4.3.1 – Message Format

The message protocol consists of a read message, a write message, and messages that can access a selected subset of the most commonly customized CobraNet MI variables. The messages all use ASCII characters to make it user-friendly and easy to use. Using ASCII also maximizes compatibility with applications that require printable ASCII characters like HyperTerminal should local PC control be desired, for example.

The messages can be characterized into the following main types:

- 1) Read variable messages.
- 2) Write variable messages.
- 3) Read response messages.
- 4) Write response messages.
- 5) Baud rate change message.

The read and write message formats follow the same basic structure. A read or write command is given first, followed by the CobraNet MI variable data. Commands and each subsequent data field must be separated by a space (designated by its ASCII hex value of 0x20). The message is completed with a carriage return (ASCII character 0x0A or <CR>) or a line feed (ASCII character 0x0D or <LF>) or both. Only a single value can be read or written at a time.

The variable being written or read can be referenced in two different ways. The first is to use the CobraNet MI variable name. The second method uses the variable's address as an ASCII hexadecimal number (without the "0x" notation) such as 100400.

A list of the variable names supported can be found in Table 13. The list also gives the variables hexadecimal addresses as well. Both methods are acceptable and both are shown in the following examples. However, it should be noted that only the most commonly used CobraNet MI variables are supported by name. If access to a non-supported variable is required, the address method must be used. All the address locations can be found in earlier sections of this document in the "Host Address" field of the CobraNet MI variable table. There are a number of different data types when writing or reading data. The table below lists the data formats and the format in which they are read or written.

Data type	Format
String	"ABCDEF"
Integer	"33"
PhysAddress (MAC Address)	"01:02:03:04:05:06"
IPAddress	"1.2.3.4"

Table 2 – Serial Message Data Types

4.3.2 – Read Messages

The general read message structure conforms to the format specified in Table 3 below.

Message Command		CobraNet MI Variable	Message Terminator
ReadVar	0x20	sysLocation	0x0D 0x0A
ReadAddr	0x20	100400	0x0D 0x0A

Table 3 – Read Request Message Format

Example: ReadVar sysLocation<CR> or ReadAddr 100400<CR>

If the variable name is used, additional parameters may be required, depending on the variable being read. For instance, the variable may require a receiver number, as used by the rxBundle variable, or a receiver number and an audio channel number, as used by the rxSubMap variable. These parameters are simply added to the message after the variable name separated by a space.

Example: ReadVar rxSubMap 1 2<CR>

The order the parameters appear in is vitally important if a variable has more than one parameter. Information on which variables are accessible and which parameters are needed for each can be found in the variable's details section.

If the variable address is used, it is the only value required as each parameter of a CobraNet MI variable has its own address. Note that only sysDescr, sysContac, sysName, sysLocation, and ifPhysAddress of the MIB II MI variable addresses are implemented.

4.3.2.1 – Read Message Response

If the variable is successfully read, the data is passed back to the user application. The format of the response is shown in Table 4 below.

Message Command		CobraNet MI Variable		Value	Message Terminator
ReadVarRsp	0x20	sysLocation	0x20	"telephone closet, 3 rd floor"	0x0D 0x0A
ReadAddrRsp	0x20	100400	0x20	"telephone closet, 3 rd floor"	0x0D 0x0A

Table 4 – Successful Read Response Message Formats

Example: ReadVarRsp sysLocation "telephone closet, 3rd floor"<CR> or
ReadAddrRsp 100400 "telephone closet, 3rd floor"<CR>

If the request fails for any reason, the response will be a *Nack* message. The format of the *Nack* message is shown in Table 5 below.

<i>Nack Reason Code</i>			<i>Message Terminator</i>
0-9 & A-D	0x20	Nack	0x0D 0x0A

Table 5 – Read *Nack* Message Format

Example: D Nack<CR>

The *Nack* response code gives an indication of the reason for the failure. A table of the reason codes and a description of each is shown in Table 12.

4.3.3 – Write Messages

The write message structure is very similar to the read message structure except it includes the new value contained within quotation marks. The format is specified in Table 6 below.

<i>Message Command</i>		<i>CobraNet MI Variable</i>		<i>New Value</i>	<i>Message Terminator</i>
WriteVar	0x20	sysLocation	0x20	“telephone closet, 3rd floor”	0x0D 0x0A
WriteAddr	0x20	100400	0x20	“telephone closet, 3rd floor”	0x0D 0x0A

Table 6 – Write Request Message Format

Example: WriteVar sysLocation “telephone closet, 3rd floor”<CR> or
WriteAddr 100400 “telephone closet, 3rd floor”<CR>

As with the read process, if the variable name is used, additional parameters may be required, depending on the variable being read.

Example: WriteVar rxSubMap 1 2 “33”<CR>

Only the data to be written is contained within quotation marks. Again, as with the read process, if the address is used, it is the only value required as each parameter of a CobraNet MI variable has its own address.

4.3.3.1 – Write Message Response

If the variable is successfully written, a simple *Ack* message is returned. The format of the response is shown in Table 7 below.

	<i>Message Terminator</i>
Ack	0x0D 0x0A

Table 7 – Successful Write Response Message Format

Example: Ack<CR>

If the write request fails, the response will be a *Nack* message. This message has the same format as the read *Nack* message and is repeated below.

<i>Nack</i> Reason Code			Message Terminator
0-9 & A-D	0x20	Nack	0x0D 0x0A

Table 8 – Write *Nack* Message Format

Example: D Nack<CR>

The *Nack* response code gives an indication of the reason for the failure. A table of the reason codes and a description of each is shown in Table 12.

4.3.4 – Baud Rate Change Message

Use this message when the baud rate of the device needs to be changed. When the baud rate change message is received by the CDK-8DUART, the change to the new baud rate will only occur if both of the following conditions are met:

1. The value is different from the current one
2. The value is a valid supported baud rate (9600, 19200, 38400, 57600, or 115200 bps).

Note that the baud rate value does not include a comma.

Message Command		New Baud Rate	Message Terminator
ChangeBaudRate	0x20	19200	0x0D 0x0A

Table 9 – Baud Rate Change Request Message

Example: ChangeBaudRate 19200<CR>

If the baud rate change request satisfies both conditions noted above, the *Ack* response message (see Table 10 below) will be transmitted at the original baud rate. Any further messages will then be sent at the new baud rate.

	Message Terminator
Ack	0x0D 0x0A

Table 10 – Successful Baud Rate Change Message Format

Example: Ack<CR>

If either condition 1 or 2 above are not satisfied, the *Nack* response message will be sent. No baud rate change will occur and further messages will be processed at the original baud rate

<i>Nack</i> Reason Code			Message Terminator
0-9 & A-D	0x20	Nack	0x0D 0x0A

Table 11 – Write *Nack* Message Format

Example: D Nack<CR>

4.3.5 – *Nack* Response Descriptions

<i>Nack</i> Reason Code	<i>Nack</i> Reason
0	Receive buffer overflow (send commands slower)
1	Transmit buffer overflow (unable to send response quickly enough)
2	Unsupported CobraNet Address received
3	Invalid counter type, IP address, timer ticks value, MAC address, or attempt to write to read-only value.
4	Attempt to read a write-only value
5	Unknown CobraNet Address received
6	Timeout waiting for CobraNet read response
7	Timeout waiting for CobraNet write response
8 - 9	Reserved
A - C	Reserved
D	Invalid Command Variable or Address
E - Z	Reserved

Table 12 – *Nack* Reason Codes

4.4 – Commands & Responses

The SHMI serial protocol processes all CobraNet MI variables by their address. However, some are addressable by name as well. The following list shows those variables. The variable definitions for these variables can be found in sections 4.4.2 to 4.4.30 below. Only the 32-bit format definitions are to be used (as opposed to the available 24-bit format).

Variable Name	Type	CobraNet Address (hexadecimal)
sysDescr	Read/Write	100000
sysContact	Read/Write	100200
sysName	Read/Write	100300
sysLocation	Read/Write	100400
ifPhysAddress	Read	11000D
flashPersistEnable	Read/Write	1100
errorCode	Read	2002
errorCount	Read	2003
modeRateControl	Read/Write	2100
conductorStatus	Read	11000
serialFormat	Read/Write	24000
serialBaud	Read/Write	24001
serialRxMAC	Read/Write	24003
serialTxMAC	Read/Write	24100
rxSubFormat	Read	4n30m (see section 4.4.16)
rxBundle	Read/Write	4n100 (see section 4.4.17)
rxSubMap	Read/Write	4n20m (see section 4.4.18)
txBundle	Read/Write	5n100 (see section 4.4.19)
txSubCount	Read/Write	5n105 (see section 4.4.20)
txUnicastMode	Read/Write	5n107 (see section 4.4.21)
txMaxUnicast	Read/Write	5n108 (see section 4.4.22)
txSubMap	Read/Write	5n20m (see section 4.4.23)
txSubFormat	Read/Write	5n30m (see section 4.4.24)
ipMonCurrentIP	Read/Write	72000
stdUserVersionMajor	Read/Write	7E000
stdUserVersionMinor	Read/Write	7E001
stdUserString	Read/Write	7E100, 7E116, 7E12C, 7E142 (see section 4.4.28)
stdUserInteger	Read/Write	7E300 – 7E340 (see section 4.4.29)
sysMAC	Read/Write	7F000

Table 13 – CobraNet MI Variable Names and Their Addresses

4.4.1 – Legend

Variable Name	Name of variable
Description	Description of the variable including allowed values and usage discussion.
Host Address	HMI addresses are used to access variables via the host port.
Parameter	Only visible for variables that require one or more parameters when reading or writing using the variable name. It lists the parameters that are needed and the order they need to be in.
Size	Size is indicated for String <i>variables only</i> . It shows the number of characters available for that string.
Type	The data type of the variable. The following type Integer: a numeric value String: An ASCII string PhysAddress: A specifically formatted string to show the MAC address. Takes the form of "aa:bb:cc:dd:ee:ff" where aa, bb, cc, dd, ee, and ff are 2-digit hexadecimal values in the range 00 to FF. IP Address: A specifically formatted string to show the IP address. Takes the form of "a.b.c.d" where a,b,c, and d are decimal values between 1 and 254.
Attributes	Read-only variables can only be read and can not be modified. Read/Write variables can be read and written. Read/Write - Persistent variables can be read and written. If the persistence feature is enabled, values of these variables will automatically be written to flash for recall at startup.
Default	Value assigned to the variable at startup when persistence is disabled. The values of some Read-only variables reflect system conditions and thus may not have a default value.
Implemented Version	Firmware version in which the variable was first introduced. Unless otherwise noted in this field, one can assume variables will be available in the version indicated and all subsequent versions.

4.4.2 – sysDescr

Variable Name	sysDescr
Description	Describes type of interface as ASCII text. Format: <product-specific description> CobraNet LE version <protocol version>.<major version>.<minor version> <hardware platform> where <hardware platform> identifies the specific IC the CobraNet LE is using. Example: Attero Tech InBox X2 CobraNet LE version 2.1.3 LM3S6918
Host Address	100000
Size	75 characters (read) / 40 characters (write) – See below
Type	String
Attributes	SNMP: Read-only SHMI: Read/Write – Persistent – See below
Default Value	<product specific description> CobraNet LE version <protocol version>.<major version>.<minor version> <hardware platform>
Implemented Version	2.1.2 – The ability to modify <product specific description> and the adding of additional version information when reading was included in 2.1.3

The default text for <product-specific description> is "Attero Tech". Any value written to *sysDescr* is immediately used in place of the default value. The maximum size for <product-specific description> is 40 characters and should take the form [Manufacturer Name] [Product Name]. This value will remain persistent regardless of how the *flashPersistEnable* parameter is set.

4.4.3 – sysContact

Variable Name	sysContact
Description	Contact details for the person responsible for this node.
Host Address	100200
Size	60 characters
Type	String
Attributes	Read/Write - Persistent
Default Value	Zero length string
Implemented Version	2.1.2

4.4.4 – sysName

Variable Name	sysName
Description	A name assigned to this managed node. By convention, this is the node's fully qualified domain name.
Host Address	100300
Size	60 characters
Type	String
Attributes	Read/Write - Persistent
Default Value	Product specific
Implemented Version	2.1.2

4.4.5 – sysLocation

Variable Name	sysLocation
Description	The physical location of this node (e.g., "telephone closet, 3rd floor")
Host Address	100400
Size	60 characters
Type	String
Attributes	Read/Write - Persistent
Default Value	Zero length string
Implemented Version	2.1.2

4.4.6 – ifPhysAddress

Variable Name	ifPhysAddress
Description	The interface's address at the protocol layer immediately 'below' the network layer in the protocol stack.
Host Address	11000D
Type	PhysAddress
Attributes	Read-only
Default Value	N.A.
Implemented Version	2.1.2

4.4.7 – flashPersistEnable

Variable Name	flashPersistEnable
Description	Non-zero value enables variable persistence feature. Read/Write - Persistent type variables will be automatically written to non-volatile memory when changed. Values will be restored on power-up.
Host Address	1100
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

4.4.8 – errorCode

Variable Name	errorCode
Description	Last 4 error codes reported. This is implemented as a 4 byte FIFO with the most recent error in the lower byte. If the same error occurs multiple times in a row it will only be put into the FIFO once.
Host Address	2002
Type	Integer
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

4.4.9 – errorCount

Variable Name	errorCount
Description	Number of errors reported during since system up time
Host Address	2003
Type	Integer
Attributes	Read-only
Default Value	0
Implemented Version	2.1.2

4.4.10 - modeRateControl

Variable Name	modeRateControl
Description	Selects latency and sample rate mode for the interface. The following modes are supported: 0x600 - 5-1/3 ms latency, 48 kHz sample rate. This was a Read/Write - Persistent variable, but CobraNet LE only supports one value so it was changed to Read-only.
Host Address	2100
Type	Integer
Attributes	Read-only
Default Value	0x600
Implemented Version	2.1.2

4.4.11 - conductorStatus

Variable Name	conductorStatus
Description	Conductor status: 0 - This interface is not the conductor 1 - This interface is the conductor
Host Address	11000
Type	Integer
Attributes	Read-only
Implemented Version	2.1.2

4.4.12 - serialFormat

Variable Name	serialFormat
Description	This variable is used to enable or disable the Serial Communications Interface. The data format for both transmit and receive directions is 8 data bits, no parity, and 1 stop bit. 0x01 - Enable serial bridging.
Host Address	24000
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

4.4.13 – serialBaud

Variable Name	serialBaud
Description	Baud rate for transmission and reception. The baud rate is specified in bits per second. The minimum baud rate is 600 baud. Maximum baud rate is 57,600.
Host Address	24001
Type	Integer
Attributes	Read/Write - Persistent
Default Value	19200
Implemented Version	2.1.2

4.4.14 – serialRxMAC

Variable Name	serialRxMAC
Description	MAC address of the CobraNet Interface from which SCI data will be accepted. This may be any multicast address, though 01:60:2B:FD:00:00 through 01:60:2B:FD:FF:FF have been reserved by Cirrus Logic for use as "asynchronous global channels". IfPhysAddress is the only usable unicast address (CobraNet does not support Ethernet promiscuous mode).
Host Address	24003
Type	PhysAddress
Attributes	Read/Write - Persistent
Default Value	01:F0:2B:FD:00:00
Implemented Version	2.1.2

4.4.15 – serialTxMAC

Variable Name	serialTxMAC
Description	MAC address of the CobraNet interface to which serial data is sent. May be any multicast or unicast address.
Host Address	24100
Type	PhysAddress
Attributes	Read/Write - Persistent
Default Value	01:F0:2B:FD:00:00
Implemented Version	2.1.2

4.4.16 – rxSubFormat

Variable Name	rxSubFormat			
Description	<p>Vector of received audio format for each sub-channel.</p> <p>The least significant bit of these variables is set when the received format is supported for reception by the CobraNet interface. A test of this least significant bit can be used to determine correct reception on a per audio channel basis.</p> <p>All entries in this vector will be 0 if rxStatus is zero.</p>			
	Audio Format Value (decimal)	Resolution	Sample Rate	Latency
	0	No Signal		
	278,528 (0x44000)	16 bit	48 kHz	5 1/3 ms
	344,064 (0x54000)	20 bit	48 kHz	5 1/3 ms
	409,600 (0x66000)	24 bit	48 kHz	5 1/3 ms
Host Address	4n30m (n is 0 based receiver number, m is 0 based audio channel number)			
Parameters	1 – Receiver number (0-7) 2 – Audio sub-channel number (0-7)			
Type	Integer			
Attributes	Read-only			
Implemented Version	2.1.2			

4.4.17 – rxBundle

Variable Name	rxBundle
Description	Receive bundle assignment.
Host Address	4n100 (n is 0 based receiver number)
Parameters	1 – Receiver number (0-7)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

4.4.18 – rxSubMap

Variable Name	rxSubMap
Description	Audio routing channel destinations for each audio channel in a received bundle.
Host Address	4n20m (n is 0 based receiver number, m is 0 based audio channel number)
Parameters	1 – Receiver number (0-7) 2 – Audio sub-channel number (0-7)
Count	8
Type	Integer
Attributes	Read/Write - Persistent
Default Value	
Implemented Version	2.1.2

4.4.19 – txBundle

Variable Name	txBundle
Description	Transmitter bundle assignment.
Host Address	5n100 (n is 0 based transmitter number)
Parameters	1 – Transmitter number (0-3)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0
Implemented Version	2.1.2

4.4.20 – txSubCount

Variable Name	txSubCount
Description	Number of audio channels to transmit in a bundle.
Host Address	5n105 (n is 0 based transmitter number)
Parameters	1 – Transmitter number (0-3)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	8
Implemented Version	2.1.2

4.4.21 – txUnicastMode

Variable Name	txUnicastMode
Description	<p>Specifies the number of unicast destinations served before automatically switching to multicast bundle transmission.</p> <p>0 - Multicast addressing used at all times. Note: multicast bundles do not transmit data until a receiver is assigned to the same bundle number.</p> <p>1 - 4 - Multicast addressing used to specify number of receivers.</p> <p>0x7FFFFFFF - Multicast addressing is never used. Maximum number of unicast destinations is set by <i>txMaxUnicast</i>. Receiver request priority is used to determine which receivers are serviced if multiple receivers are assigned to this bundle.</p>
Host Address	5n107 (n is 0 based transmitter number)
Parameters	1 - Transmitter number (0-3)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	0x7FFFFFFF
Implemented Version	2.1.2

4.4.22 – txMaxUnicast

Variable Name	txMaxUnicast
Description	<p>Specifies maximum number of unicast destinations supported simultaneously by the transmitter. Receivers in excess of this setting will not receive the bundle.</p> <p>A transmitter can service up to 4 receivers. The number of unicast destinations transmitted to will never exceed this internal capacity limitation.</p> <p>If txUnicastMode is set lower than txMaxUnicast, the bundle will switch to multicast before the limitation on unicast destinations is reached.</p> <p>If txUnicastMode is set equal to txMaxUnicast, the bundle will switch to multicast when the limitation on unicast destinations is exceeded.</p>
Host Address	5n108 (n is 0 based transmitter number)
Parameters	1 - Transmitter number (0-3)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	1
Implemented Version	2.1.2

4.4.23 – txSubMap

Variable Name	txSubMap
Description	Transmit audio channel (channel within bundle) to audio routing channel (channel of SSI) mapping. This vector contains the routing channel source specifiers per audio channel in the transmitted bundle.
Host Address	5n20m (n is 0 based transmitter number, m is 0 based sub-channel number)
Parameters	1 - Transmitter number (0-3) 2 - Audio sub-channel number (0-7)
Type	Integer
Attributes	Read/Write - Persistent
Default Value	First transmitter {1, 2, 3, 4, 5, 6, 7, 8} Second transmitter {1, 2, 3, 4, 5, 6, 7, 8}
Implemented Version	2.1.2

4.4.24 – txSubFormat

Variable Name	txSubFormat			
Description	Specifies data format for each sub-channel in the transmitted bundle. modeRateControl must also be set correctly to support the configured format.			
	Audio Format Value (decimal)	Resolution	Sample Rate	Latency
	0	No Signal		
	278,528 (0x44000)	16 bit	48 kHz	5 1/3 ms
	344,064 (0x54000)	20 bit	48 kHz	5 1/3 ms
Host Address	5n30m (n is 0 based transmitter number, m is 0 based sub-channel number)			
Parameters	1 - Transmitter number (0-3) 2 - Audio sub-channel number (0-7)			
Type	Integer			
Attributes	Read/Write - Persistent			
Default Value	0x54000			
Implemented Version	2.1.2			

4.4.25 – ipMonCurrentIP

Variable Name	ipMonCurrentIP
Description	The current IP address for the CobraNet interface. Changing the current IP address has an immediate affect on IP communications. A value of 0.0.0.0 indicates no IP address assignment for the interface. An IP address can be assigned (or reassigned) to the interface by any of the following means: A value loaded from <i>ipMonStaticIP</i> during power-up. A host processor writing to <i>pMonCurrentIP</i> via the SHMI. Receipt of a BOOTP response packet (typically in response to a transmitted BOOTP request) Receipt of a RARP response packet (RARP requests are not transmitted)
Host Address	72000
Type	IP Address
Attributes	Read/Write
Default Value	ipMonStaticIP
Implemented Version	2.1.2

4.4.26 – stdUserVersionMajor

Variable Name	stdUserVersionMajor
Description	Major user version number.
Host Address	7E000
Type	Integer
Attributes	SNMP: Read-only SHMI: Read/Write – Persistent
Default Value	0
Implemented Version	2.1.2

4.4.27 – stdUserVersionMinor

Variable Name	stdUserVersionMinor
Description	Minor user version number.
Host Address	7E001
Type	Integer
Attributes	SNMP: Read-only SHMI: Read/Write – Persistent
Default Value	0
Implemented Version	2.1.2

4.4.28 – stdUserString

Variable Name	stdUserString
Description	4 string variables to store the users string parameters.
Host Address	7E100 = User String 0 7E116 = User String 1 7E12C = User String 2 7E142 = User String 3
Parameter	1 – String number (0-3)
Size	60 characters
Type	String
Attributes	Read/Write - First 2 strings are Persistent
Default Value	Zero length string
Implemented Version	2.1.2

4.4.29 – stdUserInteger

Variable Name	stdUserInteger
Description	64 integer variables to store the user's integer parameters.
Host Address	7E300 through 7E340
Parameter	1 – Integer Number (0-64)
Type	Integer
Attributes	Read/Write - First 32 are Persistent
Default Value	0
Implemented Version	2.1.2

4.4.30– sysMAC

Variable Name	sysMAC
Description	The system MAC variable was added to allow setting the MAC address of the CobraNet LE board. It should be set once during production. The address is saved at the last 6 bytes of Persistent memory
Host Address	7F000
Type	PhysAddress
Attributes	Read/Write – Persistent – See below
Default Value	FF:FF:FF:FF:FF:FF
Implemented Version	2.1.2

This value will remain persistent regardless of how the flashPersistEnable parameter is set.

5 – Error Code Reference

5.1 – Legend

Byte Code - Numeric error code. Error codes reported through SNMP or SHMI are of varying form must be converted to this common byte code representation as per instructions given in section 5.2 below.

Type - Classification and behavior of the error condition.

Type	Description
TX	Recoverable and expected transmit error
TXQUIET	Informative transmit incident
TXFAULT	Unexpected but recoverable transmit error
RX	Recoverable and expected receive error
RXQUIET	Informative receive incident
RXFAULT	Unexpected but recoverable receive error
TXRX	Recoverable and expected error simultaneously affecting transmit and receive
FAULT	Unexpected but recoverable error
FATAL	Unrecoverable error condition.

Table 14 - Error Types

Name - Name assigned to the error by the firmware programmer.

Description - Description of the error condition.

Expected Conditions - Foreseeable conditions under which the error condition would occur on normally functioning and properly connected hardware.

Unexpected Conditions - Conditions indicating a hardware or firmware fault.

5.2 – Error Code Interpretation

Errors are listed and described in section 5.3.

A conversion between raw error codes reported via the errorCode MI variable and the byte code key values is required to correctly interpret error conditions. The conversion between error code and byte code is shown below.

The errorCode MI is a 4 byte FIFO with the most recent error byte code in the lower 8 bits. If an error occurs multiple times in a row it will only be put into the FIFO once. To retrieve the most recent byte code, strip off the most-significant 24 bits by taking the error code modulo 256. Alternatively, when the error code is expressed as a hexadecimal value, the byte code (in hexadecimal form) appears in the first and second digits. Convert this value to decimal representation and look up the byte code in the table below.

5.3 – Error Codes Listing

Byte Code	Type	Name	Description	Expected Conditions	Unexpected Conditions
0	NONE	NO_ERROR	No error has been reported.	-	-
1	FATAL	NO_CODE	No valid runtime code to load from FLASH memory.	No code has ever been loaded in the FLASH.	Corrupted FLASH contents. Hardware failure in flash memory.
2	FATAL	BOOT_CSUM	Bad boot sector checksum.	-	Corrupted FLASH contents. Hardware failure in flash memory.
16 0x10	FAULT	FLASH_FAILURE	Failure during FLASH erase or write operation.	Internal FLASH device has fatigued due to excessive write/erase operations.	Problem with flash device.
18 0x12	FATAL	BAD_CONFIG	Bad configuration record in EEPROM	External EEPROM device has fatigued due to excessive write/erase operations.	There is a problem with the I2C interface to the external EEPROM.
22 0x16	FATAL	POST_CLOCK_RANGE	Sample clock pull range test failure. The voltage controlled sample clock crystal oscillator (VCXO) pull range does not meet minimum requirements.	-	The VCXO device does not meet specification. Problem with VCXO control voltage circuitry.
23 0x17	FATAL	POST_CLOCK_STOPPED	Sample clock not running. Timeout waiting for measurement edge.	-	VCXO is not oscillating. Problem with VCXO circuitry or CPLD.
36 0x24	TX	TX_TOOBIG	Attempt to generate an oversize outgoing packet.	Too many 24 bit audio channels specified for transmitter.	-
39 0x27	TXRX	LOST_LOCK	Lost lock to network clock.	Initial synchronization to the network. Change of conductor. Conductor is attempting to synchronize to bad external reference clock.	Sample clock crystal of either this unit or the conductor does not meet specification.
40 0x28	RX	EARLY_PACKET	Received an audio packet with eager timestamp.	Excessive delay variation through the network.	-
42 0x2A	TXFAULT	QUEUEPUT_BEATDISCARDS	Packet queue overflow discarding untransmitted packets.	-	-
46 0x2E	RX	QUEUEPUT_RX	Too much audio data received for a network isochronous channel.	-	Two devices are mistakenly transmitting onto the same bundle.

Byte Code	Type	Name	Description	Expected Conditions	Unexpected Conditions
52 0x34	RX	RXBUFFER_OVERFLOW	Receive buffers exhausted.	Excessive broadcast traffic on network. Setting of rxMinDelay is too high. Excessive delay variation through network.	-
53 0x35	TXFAULT	SSIRX_OVERRUN	Synchronous serial (SSI) audio data receive overrun.	-	Excessive interrupt latency.
54 0x36	RXFAULT	SSITX_UNDERRUN	Synchronous serial (SSI) audio data transmit underrun.	-	Excessive interrupt latency.
57 0x39	TX	TX_CHANCOUNT	Unable to transmit all outbound audio packets within an isochronous cycle period.	Bandwidth is exhausted on a repeater network.	-
60 0x3c	TXQUIET	TXAUDIO_DROPOUT	Audio dropout occurred on transmission.	Conductor revoked permission, channel number changed or turned off by user. A beat packet did not arrive either due to a conductor change or a problem on the network.	-
61 0x3D	RXQUIET	RXAUDIO_DROPOUT	Audio dropout occurred on reception.	Conductor revoked permission, channel number changed or turned off by user. Audio packet was dropped by the network.	-
64 0x40	TX	QUEUEPUT_SCICRX	Serial bridge cannot packetize and transmit characters as fast as they are being received from the serial port.	SerialPPeriod setting too high for selected baud rate.	-
73 0x49	RX	QUEUEPUT_SCITXC	Serial bridging received more characters from the network faster than it can transmit them out the serial port.	Characters are being received simultaneously over the network from two transmitters. Baud rate at transmitter is set higher than at receiver.	-
74 0x4A	RXQUIET	RXIDLE	Loss of receive activity detected.	There are no longer any other CobraNet devices on the network.	-
79 0x4F	RXFAULT	ETHERRX_OVERRUN	Could not keep up with Ethernet receive data transfer requirements.	-	Problem with Ethernet receive DMA.
80 0x50	RX	ETHERRX_CRC	Corrupted Ethernet packet received (CRC error).	Late collisions on an oversized (>200 meter) repeater network. Cabling, equipment, or electromagnetic interference problem on the network. A cut through switch may convert collision fragments into packets with bad CRC.	Problem with Ethernet controller.

Byte Code	Type	Name	Description	Expected Conditions	Unexpected Conditions
88 0x58	TX	SCI_RXOVER	Asynchronous serial receive overrun error.	Bad serial baud rate selected.	-
90 0x5A	TX	SCI_RXFRAMING	Asynchronous serial receive framing error.	Wrong serial data format or baud rate selected.	-
114 0x72	RX	NO_BEAT_HEADER	Could not find header section in beat packet.	-	Conductor has transmitted an ill formed beat packet.
115 0x73	RX	NO_ISO_HEADER	Could not find header section in isochronous data packet.	Device has transmitted an ill formed isochronous data packet	-
121 0x79	RX	SCIDATA_MISSING	Serial bridge packet contained no data section.	-	Malformed packet transmitted by another station or packet was truncated on receipt.

6 – Notice of Open Source Software Components used in the CobraNet LE firmware

The following acknowledgements are to the authors of various sections of code used in the CobraNet LE.

Bootloader code

<http://www.menie.org/georges/embedded/tftpnaive/>

```

/*
Copyright 2001, 2002 Georges Menie (www.menie.org)

This file is part of Tftpnaive.

Tftpnaive is free software; you can redistribute it and/or modify
it under the terms of the GNU Lesser General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

Tftpnaive is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License
along with Tftpnaive; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

```

AES Encryption Algorithm

This code is based off of code under the Lesser GNU Public License. The code is located at

http://www.irisa.fr/texmex/people/dufouil/ffmpegdoxy/aes_8c-source.html

```

/*
* copyright (c) 2007 Michael Niedermayer <michaelni@gmx.at>
*
* This file is part of FFmpeg.
*
* FFmpeg is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 2.1 of the License, or (at your option) any later version.
*
* FFmpeg is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public
* License along with FFmpeg; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
*
* some optimization ideas from aes128.c by Reimar Doeffinger
*/

```

Operating System

<http://www.freertos.org/>

```

/*
    FreeRTOS.org V4.6.0 - Copyright (C) 2003-2007 Richard Barry.

    This file is part of the FreeRTOS.org distribution.

    FreeRTOS.org is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    FreeRTOS.org is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with FreeRTOS.org; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

    A special exception to the GPL can be applied should you wish to distribute
    a combined work that includes FreeRTOS.org, without being obliged to provide
    the source code for any proprietary components. See the licensing section
    of http://www.FreeRTOS.org for full details of how and when the exception
    can be applied.

    *****
    See http://www.FreeRTOS.org for documentation, latest information, license
    and contact details. Please ensure to read the configuration and relevant
    port sections of the online documentation.

    Also see http://www.SafeRTOS.com for an IEC 61508 compliant version along
    with commercial development and support options.
    *****
*/

```

TCP/IP stack

<http://savannah.nongnu.org/projects/lwip/>

```

/*
 * Copyright (c) 2001-2004 Swedish Institute of Computer Science.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
 * SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
 * OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
 * IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
 * OF SUCH DAMAGE.
*/

```

```
*
* This file is part of the lwIP TCP/IP stack.
*
* Author: Adam Dunkels <adam@sics.se>
*
* Improved by Marc Boucher <marc@mbsi.ca> and David Haas <dhaas@alum.rpi.edu>
*
*/
```

SNMP

```
/*
* Copyright (c) 2006 Axon Digital Design B.V., The Netherlands.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
*
* 1. Redistributions of source code must retain the above copyright notice,
*    this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
*    this list of conditions and the following disclaimer in the documentation
*    and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
* SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
* OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
* IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
* OF SUCH DAMAGE.
*
* Author: Christiaan Simons <christiaan.simons@axon.tv>
*/
```

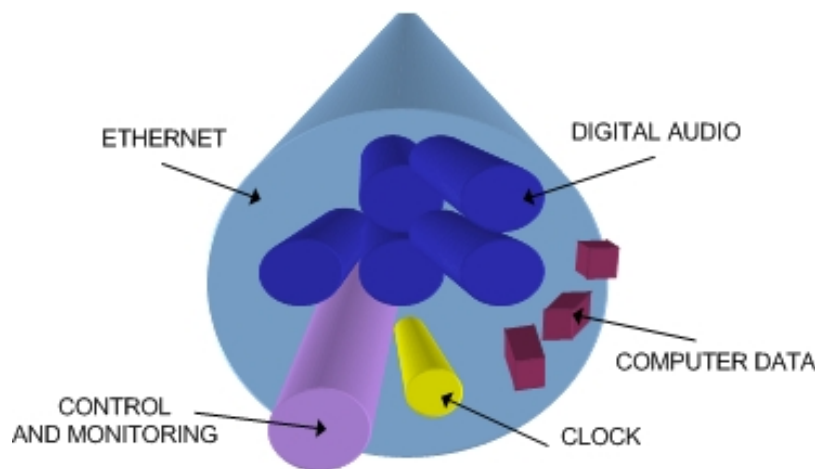
APPENDIX A – Introduction to CobraNet

CobraNet is an audio networking technology for delivery and distribution of real-time, high quality, uncompressed digital audio using a standard Ethernet network. It is implemented using a combination of hardware, firmware, and the CobraNet protocol.

Unlike other audio networking or distribution technologies, CobraNet is a true network and exists on standard Ethernet networks using standard Ethernet hardware. Since it is a true network, audio routing is highly flexible between network nodes and can be used in a variety of audio distribution applications.

In addition to the high degree of routing flexibility that CobraNet provides, the technology also incorporates the ability to monitor and control CobraNet devices remotely. This is a key feature that is highly important in fixed installation applications where the audio distribution equipment may not be readily accessible. All CobraNet devices on the network can be controlled and monitored from a central location by sending control commands and monitoring device specific parameters.

CobraNet provides this capability by implementing Simple Network Management Protocol (SNMP). SNMP is a standard protocol typically used for monitoring network devices such as Ethernet switches. In the case of CobraNet, it allows users to communicate with any CobraNet device using standard SNMP tools or a customized user interface designed specifically for CobraNet, such as Attero Tech's Control Center application.



The figure above represents the types of data that coexist on a CobraNet network.

Before a CobraNet system can be configured, it is important to first understand how CobraNet distributes audio between devices.

Audio is sent in "bundles" on a CobraNet system. Each bundle is capable of holding up to 8 logical audio channels. Every CobraNet device has a number of bundle transmitters and bundle receivers. These transmitters and receivers are the mechanism used to send and receive bundles between devices.

For a transmitted bundle, audio may be sourced from either the local audio inputs of the device or internal audio from the on-board DSP¹, but not both simultaneously. Combinations of the local or internal audio may exist within a bundle in any order. Additionally, a single source in a device may be used multiple times in a single transmitter bundle or across multiple transmitter bundles. For a received bundle, the received network audio may be routed to the device's local outputs or the internal DSP¹ or simply ignored.

Once the contents of a bundle have been decided, the next step is to pass to another CobraNet device. To do this, every CobraNet device has up to 4 bundle transmitters. Each bundle transmitter has a transmit mode that must first be selected. This affects how many devices may receive that particular bundle at a time. The modes are as follows:

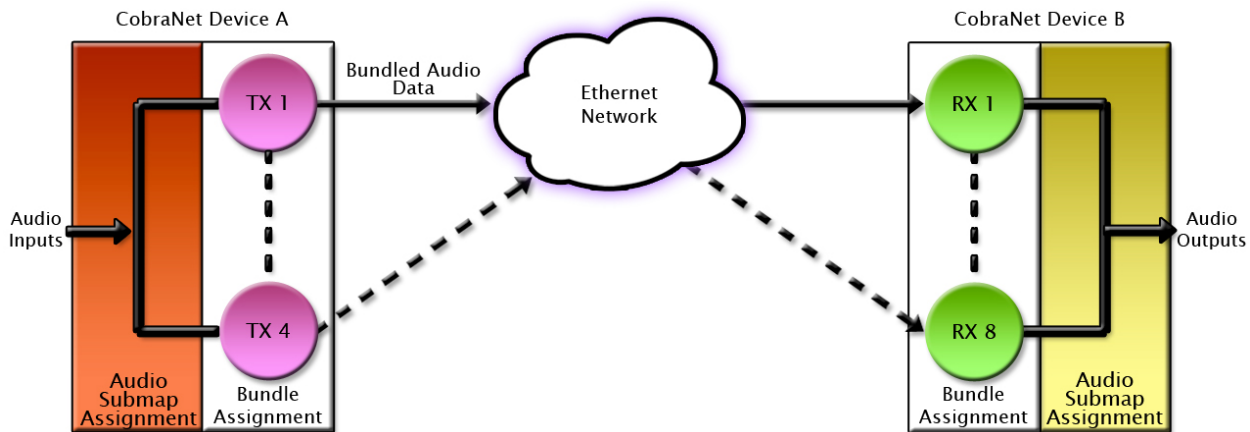
- Unicast – Used for one-to-one connections. In this mode, only one receiver at a time can receive this bundle. Once a link is established from this transmitted bundle to a receiver, any future requests for that bundle will fail.

¹ Not available on all devices – CS496xxx devices only

- Multicast – Used for one-to-many connections. This mode broadcasts its contents over the entire network. There is no restriction on the number of receivers. However, the downside is the network bandwidth required.
- Multi-unicasts – Another one-to-many mode. Whilst this is the most efficient method for getting a bundle to multiple receivers in terms of network bandwidth, it requires more processing power on the CobraNet device so in this mode there is a maximum limit of for receiver connections (this can be reduced if required). If more connections are required than the limit, the node automatically switches to multicast.

Note: When a bundle must be transmitted to multiple receivers, multi-unicast transmissions should be used where possible.

Once the mode is selected, to enable a device to transmit the bundle, simply allocate the particular transmitter bundle a non-zero number. Since this number identifies all the network packets sent out by that transmitter, each transmit bundle number must be unique on a network².



Now that the transmitter is set up, it is time to set up the receivers. In order to receive bundles, each CobraNet device has up to eight bundle receivers. To enable a device to receive a bundle, simply allocate one of that device's bundle receivers the same bundle number as a transmitted bundle. By doing so, a virtual link is created and audio should now be passed from one device to the other. It should be noted that no knowledge of a device's network topology or connection is thus required in order to configure audio connections. The only restriction to this is that a device cannot be set up to receive a bundle it is also transmitting.

The above case creates a simple, one-to-one, unidirectional link. If more devices are required to receive that bundle, allocate the same transmitted bundle number to a bundle receiver on the other CobraNet devices.

It is also important to note that CobraNet supports simultaneous bidirectional audio distribution in each device. Not only could audio be sent from Device A to Device B but at the same time, should it be needed, audio could also be sent from Device B to Device A. The exact bundle and routing configuration will be determined by the needs of each individual installation. An installation may have multiple units transmitting multiple bundles. The only restriction is the bandwidth available on the network to transfer the audio.

CobraNet does more than just transfer audio data. It can be used to pass serial information as well. A feature called serial bridging has been incorporated that allows the passage of serial data between nodes. Each node can pass serial data to a specific node or multicast the data to multiple nodes. A node can also receive data from either a single source or multiple sources. Baud rates, data bits, stop bits, parity, and so on are all configurable. There is also support for multi-drop serial buses as well.

Finally, CobraNet has the capability to alter all of the above options in real time making the whole system completely dynamic. By use of control software, all of the bundle assignment parameters can be configured with no need to change cables, switch out connectors, or pull new wiring. Most importantly, this control capability can be implemented from a single location!

² Bundle numbers range from 1 through 65535. A value of 0 represents an inactive bundle. Numbers 1-255 are reserved for multicast mode transmissions.

APPENDIX B - Reference Documents

The following table lists the relevant reference documents.

Document Title
CobraNet Programmer's Reference (Cirrus Logic)