Domosys Corporation

CEWay™ PL-III

CEWay™ is a family of chips developed by DOMOSYS Corp. to meet the requirements of the residential and commercial local area networks (LANs). The CEWay PL-III is the ideal device for mid- to high-end electronic appliances, such as central controllers. It integrates the complete Physical and Data Link Layers of the CEBus® standard (EIA-600). The designer must add a host processor to perform the functions of the upper layers of the protocol and run the user application. The CEWay PL-III is designed for superior performance in noisy power line environments.

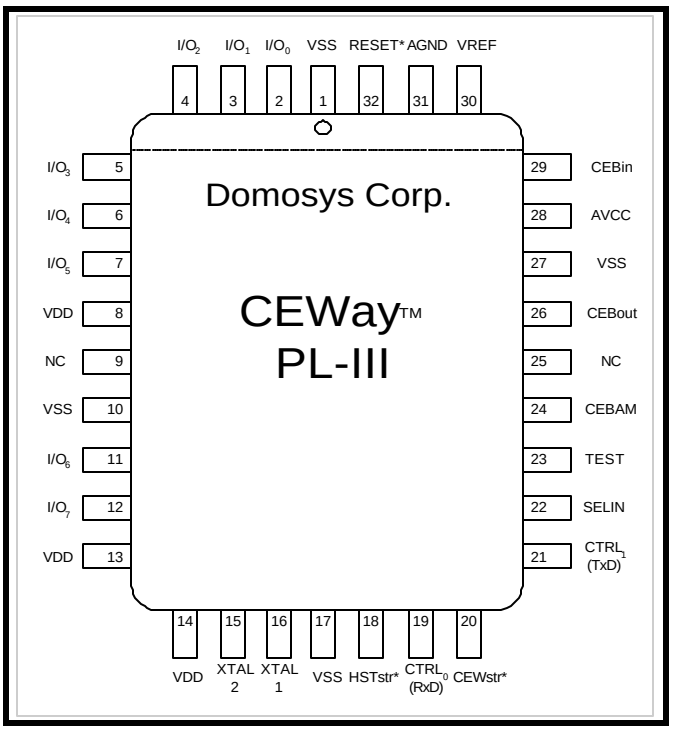


Figure 1 Pinout for CEWay PL-III

Features

- CEBus power line Physical Layer
- Power line medium dependent Physical and Symbol Encoding Sublayers
- CEBus Data Link Layer
- Proprietary DSP for superior signal reception in noisy environments
- Acknowledged and Unacknowledged services with or without addressed services
- Up to 27 group addresses
- Parallel or serial interface to host
- 32-pin PLCC Package
- Industrial operating temperature range

CEWay PL-III Block Diagram

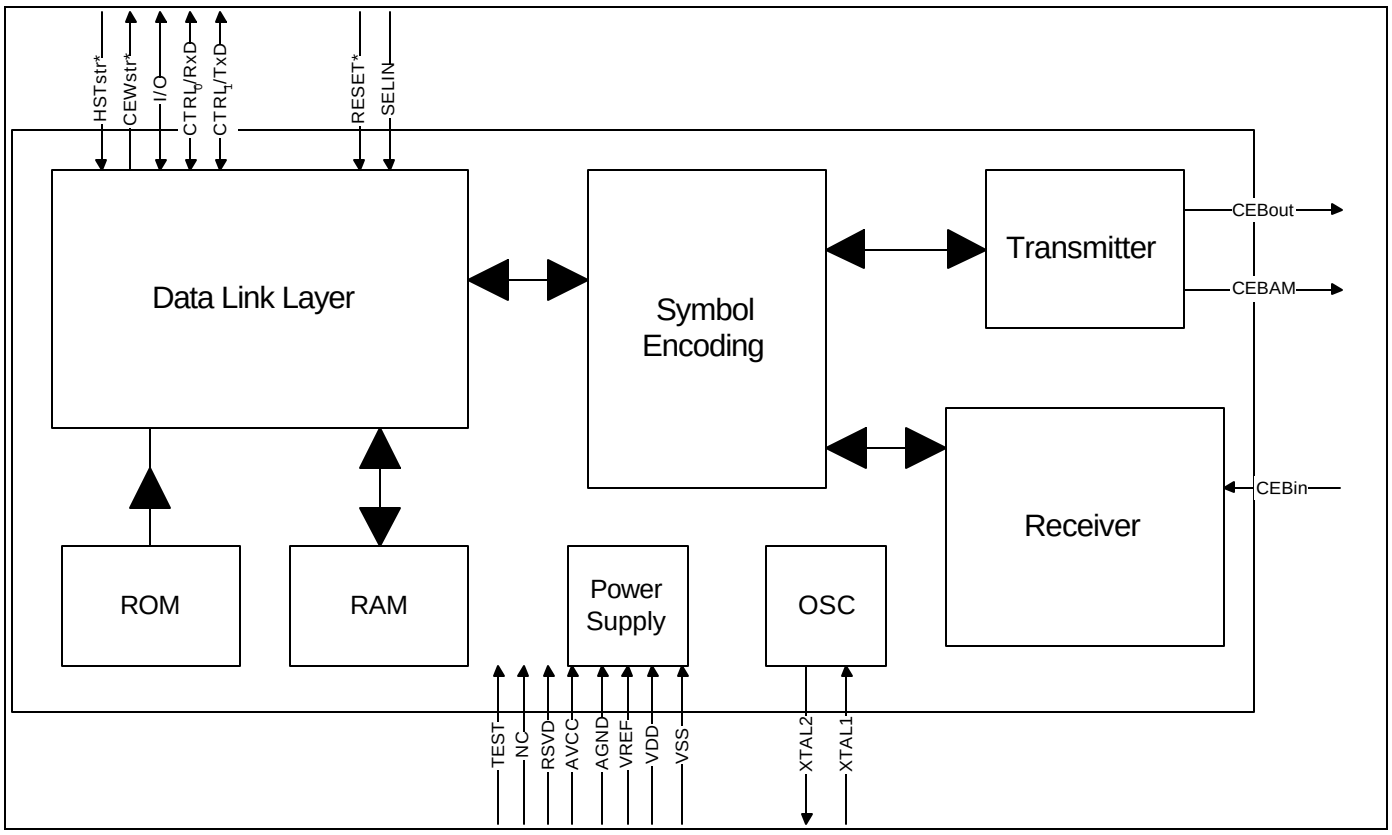


Figure 2 CEWay PL-III Block Diagram

Pin Descriptions

Table 1 CEWay PL-III Pin Descriptions

Pin	Name	Description
8,13,14	**VDD**	Power Supply: Digital 5 volts.
1,10, 17,27	**VSS**	Ground: 0 volt Digital Reference.
28	**AVCC**	Power Supply: Analog 5 volts.
31	**AGND**	Ground: 0 volt Analog Reference.
32	**RESET***	Reset: A LOW on this pin for 24 clock cycles (1.68 µs) while the oscillator is running resets the device. This pin has an internal pull-up resistor.
2,3,4,5,6, 7,11, 12	**I/O [0 - 7] (BR)**	I/O is an 8-bit bi-directional port. This port is used with the parallel host interface as the data port. When operating in the serial mode, it is used to configure the baud rate (BR) of the serial host interface. **Note: Some designs may require pull-up resistors on these pins. 4.7 kW should be sufficient in most cases.**
20	**CEWstr***	CEWay Strobe is an output bit used in the parallel communication with the host. **Note: Some designs may require a pull-up resistor on this pin. 4.7 kW should be sufficient in most cases.**
18	**HSTstr***	Host Strobe is an input bit used in the parallel communication with the CEWay PL-III.
19	**CTRL0/RxD**	Control Status Line 0: Bi-directional pin for control information when using the parallel host interface. Receive pin for the serial host interface. **Note: Some designs may require a pull-up resistor on this pin. 4.7 kW should be sufficient in most cases.**
21	**CTRL1/TxD**	Control Status Line 1: Bi-directional pin for control information when using the parallel host interface. Transmit pin for the serial host interface. **Note: Some designs may require a pull-up resistor on this pin. 4.7 kW should be sufficient in most cases.**
26	**CEBout**	CEBus power line analog signal output. 0.5V to 4.0V.
24	**CEBAM**	Digital output used to enable and disable the transmit amplifier.
29	**CEBin**	CEBus power line analog signal input.
16	**Xtal1**	Crystal 1: Input to the inverting oscillator amplifier that forms the oscillator.
15	**Xtal2**	Crystal 2: Output to the inverting oscillator amplifier that forms the oscillator.
22	**SELIN**	Select Host Interface: This input pin selects the interface to be used for the communication with the host processor. A HIGH applied to this pin selects the parallel interface configuration. A LOW applied to this pin selects the serial interface configuration. This pin has an internal pull-up resistor.
30	**VREF**	Input for analog reference. A 1 µF capacitance must be put between this pin and AGND.
23	**TEST**	TEST: Must be connected to VSS.
9, 25	**NC**	No Connect: These pins should not be connected.

Electrical Specifications

Absolute Maximum Rating[1]

Table 2 CEWay PL-III Maximum Ratings

Parameter	Sym	Min	Max	Units	Test Conditions
Supply Voltage	V_{DD}- V_{ss}	-0.3	7	V	
DC Input Voltage	V_{IN}	-0.3	VDD +0.3	V	
DC Input Current	I_{IN}	-10	+10	mA	
Storage Temperature	T_{STG}	-40	+125	^{O}C	
ESD Tolerance			2	kV	
Power Dissipation (package)	P_{DISS}			W	

[1] Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.

DC Electrical Characteristics - Voltages are with respect to VSS unless stated otherwise.

Table 3 CEWay PL-III DC Electrical Characteristics

Parameter	Sym	Min	Typ[2-3]	Max	Units	Test Conditions
Supply Voltage - Digital	VDD	4.75	5	5.25	V	
Supply Voltage - Analog	AVCC	4.75	5	5.25	V	
Input Voltage (high)	V_{IH}	0.7VDD		VDD+0.3	V	
Input Voltage (low)	V_{IL}	VSS-0.3		0.3VDD	V	
Output Voltage (high)	V_{OH}	2.4			V	I_{OH} = 50 µA
Output Voltage (low)	V_{OL}			0.4	V	I_{OL} = 4 mA
Operating Temperature	T_O	-40		+85	^{O}C	
Operating Current Digital	I_{VDD}		25	35	mA	
Operating Current Analog	I_{AVCC}		9.5	15.5	mA	

[2] Typical figures are at 25 °C and are for design aid only: not guaranteed and not subject to production testing.

[3] DC Electrical Characteristics are over recommended temperature range & recommended power supply voltages.

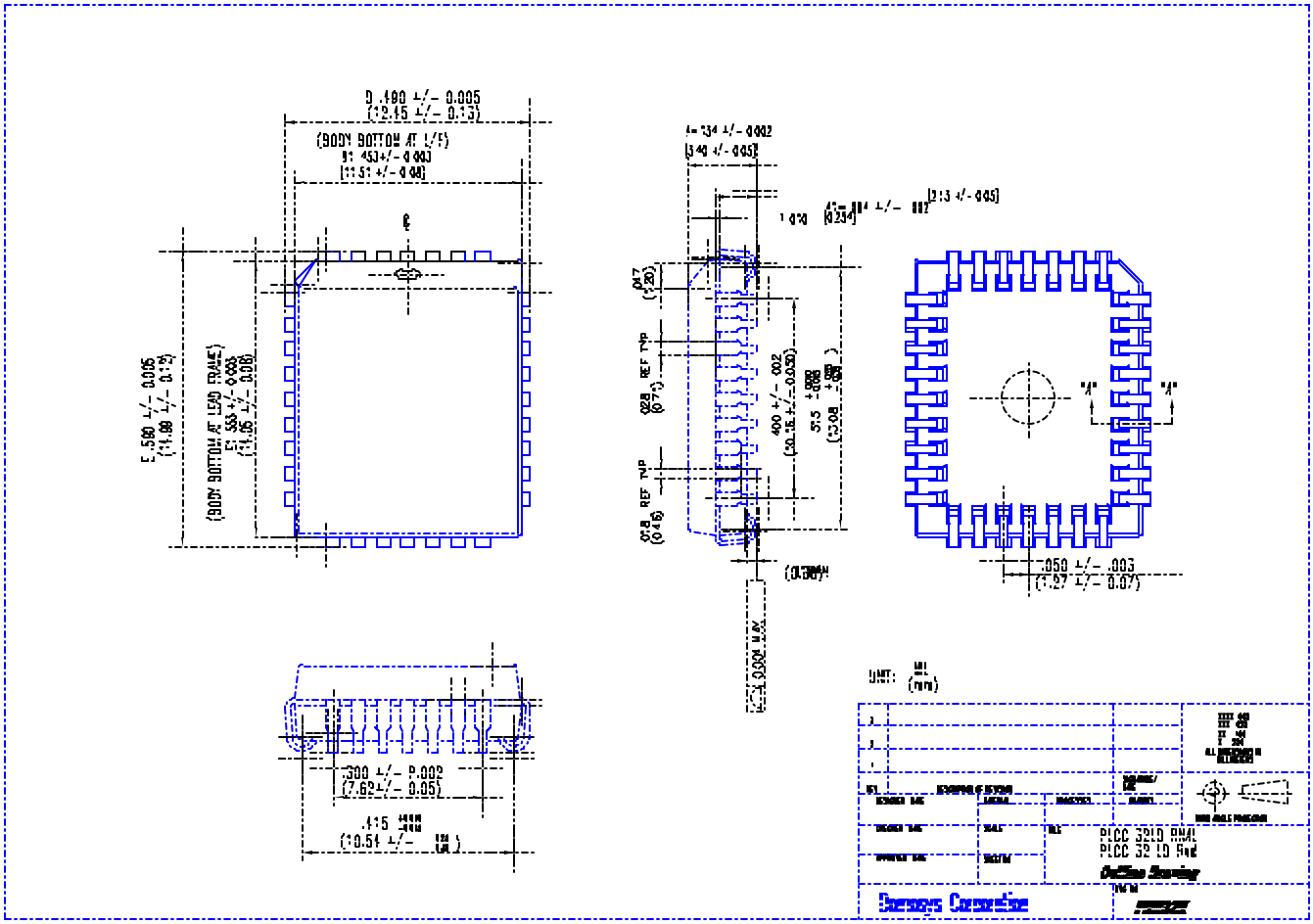
AC Electrical Characteristics

Table 4 CEWay PL-III AC Electrical Characteristics

Parameter	Sym	Min	Typ[4]	Max	Units	Test Conditions
CEBout Output Voltage	V_{CEBout}		3.5		Vp-p	
CEBout Load Impedance	ZL_{CEBout}	5			kΩ	
CEBin Input Impedance	Zin_{CEBin}	40	68	140	kΩ	
Clock Frequency	f_{CLK}		14.318		MHz	
Pin Capacitance XTAL1,XTAL2	C_{XTAL}		0.8		pF	
Input Pin Capacitance	C_I		8		pF	
Output Pin Capacitance	C_o		8		pF	

[4] Typical figures are at 25 °C and are for design aid only: not guaranteed and not subject to production testing.

Packaging



Ordering Information

Table 5 CEWay PL-III Ordering Information

Part Number	Description
PIII-000-01	CEWay PL-III Integrated Circuit

CEBus Standard Implementation

The CEWay PL-III fully implements the CEBus standard power line Physical and the Data Link Layers (PhyL and DLL). Certain parameters of these layers and of the implementation can be adjusted to meet the specific objectives of the product developer, which can be easily accomplished using Domosys' CEBox™ software tool.

Medium-Dependent Physical Sublayer & Symbol Encoding Sublayer

The transceiver (MDPS and PLSES) is composed of a transmit chain and of a receive chain. The transmitter provides an analog representation of a stored ROM code to an off-chip amplifier for transmission. Upon reception, the device samples the incoming filtered waveform through an ADC and provides a bit-by-bit comparison with an internally stored set of waveform probabilities. When a digital threshold is exceeded, the bit is stored with the rest of the message.

The symbol decoder performs leading zero suppression and recognizes the Preamble EOF string in order to force the device into the receive mode. The symbol decoder recognizes the incoming signals from the transversal filter and prepares them for storage in RAM.

The MDPS and PLSES embed a part of the Layer System Management (LSM).

Medium Access Control Sublayer

The CEBus Data Link Layer (DLL) is completely contained within its Medium Access Control Sublayer (MACS); the Logical Link Control Sublayer (LLCS) is not used in the CEBus DLL. The MACS is the only layer within the CEWay PL-III to which the host has access . All requests, indications, and confirms are sent to, or originate from the MACS. Therefore communication between the host and the PL-III is often spoken of as communication between the host and the MACS.

Features

- Supports Home Plug & Play broadcast address recognition
- Supports all configurations, parameters and features of the CEBus standard (EIA-600) Data Link Layer and Physical Layer
- Supports both Acknowledged and Unacknowledged services with the following protocol parameters :
    - CH_ACCESS_NUM
    - CH_ACCESS_PERIOD
    - RETRANS_TIME
    - MAX_RESTART
- Supports both Addressed and Non-Addressed service types
- Supports all three priority levels

- Supports either long (8 Symbol) or short (0 Symbol) Preamble Field for Iack Types (IACK, FAILURE or ADR_IACK) and Immediate Retries (IRetry)
- Supports either long (with 2 Bytes result-oriented data) or short (no data) Information Field for Non-Addressed Iack Types
- Supports up to 27 different group addresses
- Supports Addressed Packet Duplicate Rejection based on an up to ten-element duplicate list for receptions and another up to ten-element duplicate list for transmissions
- Supports higher priority requests
- The MACS embeds part of the LSM.

Host Interface

The Host Interface is the link between the CEWay PL-III and the host processor. Functions available via the Host Interface are:

- Test mode
- Initialization
- Transmission of a frame
- Reception of a frame
- Monitoring the PL-III status

The Host Interface can be configured as either a parallel or a serial interface. Table 6 shows the correspondence between the SELIN pin and the interface selected.

Table 6 Interface Selection via SELIN Pin

Interface Type	***SELIN***
Parallel	1
Serial	0

Test Mode

Irrespective of how the SELIN pin is set, the PL-III can be forced into a test mode by holding both Control lines (or Rx and Tx lines for the serial interface) low while resetting the PL-III via its RESET* pin. The PL-III will then transmit the following standard priority, addressed, unacknowledged packet for a duration of 750 ms. The Reserved bit in the control field is set to indicate that it is not a normal packet, and should normally not be received by nodes having a Unit Address of 0x0001 and House Code of 0x1000.

Table 7 Packet Transmitted in Test Mode

Control Field	DA	DHC	SA	SHC	Data
2F	0001	1000	0002	2000	50 E8 49 07 45 43 F5 31

This mode can be useful to verify that the PL-III has most of its pins properly connected, and that the analog output section is working. If no transmission occurs when both Control lines are held low and the RESET* pin is pulsed low, all hardware connections should be verified before proceeding to more complicated tests.

Parallel Interface

The CEWay PL-III uses the parallel interface to communicate with the host processor when V_{CC} is applied to the SELIN pin. The I/O port is used as a bi-directional port. CEWstr*, HSTstr* and $CTRL_{[0,1]}$ are used as control lines. The control pins are defined as follows:

Table 8 Control Pins for Parallel Interface

Mnemonic	Description	Dir. (from CEWay)
CEWstr*	CEWay Strobe	Output
HSTstr*	Host Strobe	Input
CTRL0	Control Line 0 (LSB)	Bi-directional
CTRL1	Control Line 1 (MSB)	Bi-directional

Note that some designs may require pull-up resistors on pins CEWstr*, CTRL0, CTRL1, and I/O 0-7. A 4.7 kΩ resistor will be sufficient in most cases.

There are three valid modes that the CEWay can be found in. These are the Status mode, the CEWay Communication mode and the Host Communication mode.

Status Mode

The CEWay is in the status mode when neither of the two other modes applies. The Control lines, $CTRL_0$ and $CTRL_1$, are used to indicate the status of CEWay PL-III to the host.

Table 9 Correspondence Between Control Lines and CEWay PL-III Status

CTRL1	CTRL0	STATUS
0	0	Not used
0	1	Not used
1	0	CEWay is about to send an indication to the host
1	1	CEWay is in Ready state

STATUS 00

Not used.

STATUS 01

Not used.

STATUS 10

This status means the CEWay is about to send an indication or a confirm to the host. The CEWay writes a value of 10 bin to the Control lines for 70 µs, before entering into the synchronization phase of the CEWay Communication mode. This is to allow the host enough time to back off from any request that it might be preparing.

STATUS 11

This status indicates that the CEWay is in its Ready state, and will enter the Host Communication Mode if it begins receiving a request from the host.

CEWay or Host Communication Mode

In these modes, the CEWay or the host starts a data transfer. The data transfer is performed in 3 phases.

Synchronization Phase

To proceed with the Synchronization phase, the transmitter must assert its strobe and monitor the receiver strobe. When the receiver is ready to enter in the transfer phase, it sends a negative pulse to the transmitter. The transmitter then disables its strobe to tell the receiver that it will begin the transfer. Note that the CEWay also writes a value of 10 bin to the two Control lines for 70 µs prior to setting the CEWstr low in the CEWay to Host Synchronization. The host should always check the value of the Control lines prior to sending a request to the CEWay. If the value on the Control lines is 10 bin, then the host should wait for the CEWay to send its data first. Otherwise, the host must pull the HSTstr* line low within 70 µs to start its request. Once read, the status of the Control lines is no longer important in the Host to CEWay Synchronization Phase.

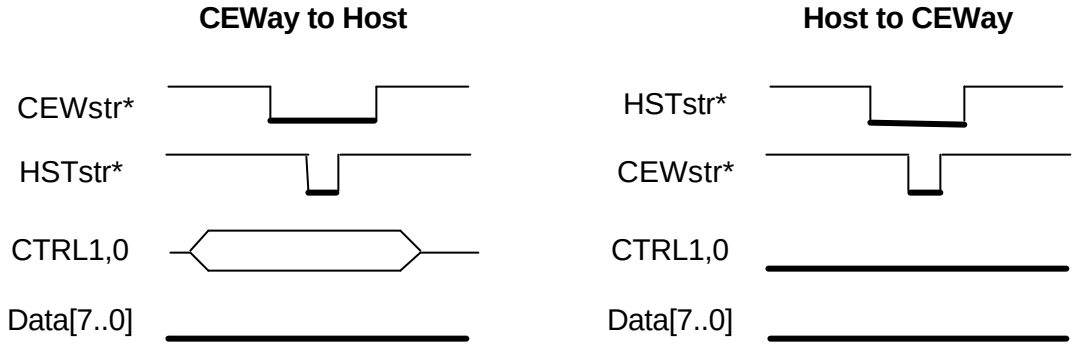


Figure 3 CEWay PL-III Synchronization Phase for Parallel Interface

Transfer Phase

The transmitter sends the message to transmit in the following format:

<Primitive ID Byte> <Data> <Checksum>

In the transfer phase, the transmitter starts by putting the corresponding Primitive Type value's 2 LSBits on the Control lines (1,0) and then puts the Primitive ID byte on the Data bus (Data 0-7). When done, the transmitter sends a negative pulse. This negative pulse indicates to the receiver that the DATA is ready for reading. When the receiver is ready for another byte, it

sends a negative pulse on its strobe line. The transmitter must maintain the Primitive Type's value on the Control lines during the whole transfer phase.

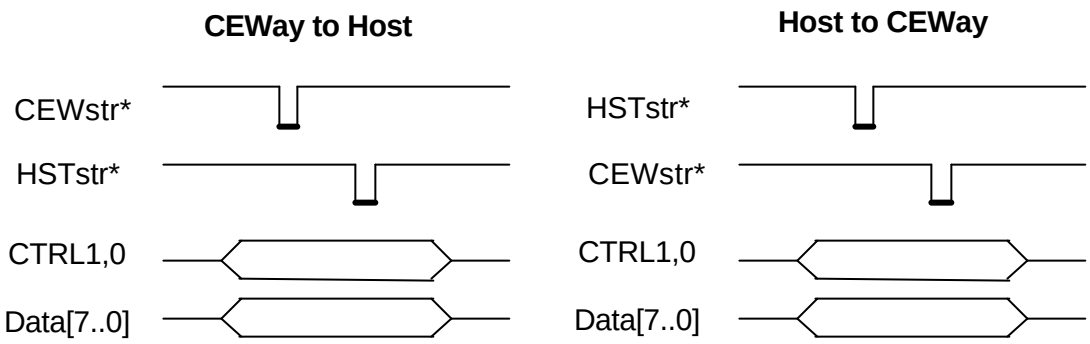


Figure 4 CEWay PL-III Data Transfer Phase for Parallel Interface

Checksum Phase

This phase tells the receiver that the next byte is the last byte of the sequence. This byte is the two's complement of the sum of all preceding bytes sent during the transfer phase. The Transmitter pulls both Control lines low, and puts the checksum on the Data bus. When done, the transmitter sends a negative pulse on its Strobe line. When the receiver has read those values, it replies with a negative pulse. The transmitter must then internally change the direction of the port from output to input. Once this is done, the transmitter replies with a negative pulse on its Strobe to indicate that it is ready to read the Checksum Status from the receiver.

When the receiver has completed the computation of the checksum, it writes the Checksum Status to the data lines, and sends a negative pulse to the transmitter.

Table 10 Checksum Response Codes for Parallel Interface

Value sent to the Transmitter	Checksum STATUS
0x00 Hex	Transfer OK
0x01 Hex	Error in transfer

CEWay to Host

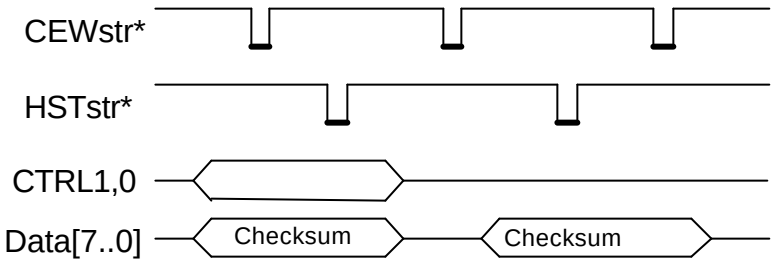


Figure 5 CEWay to Host Checksum Sequence

Host to CEWay

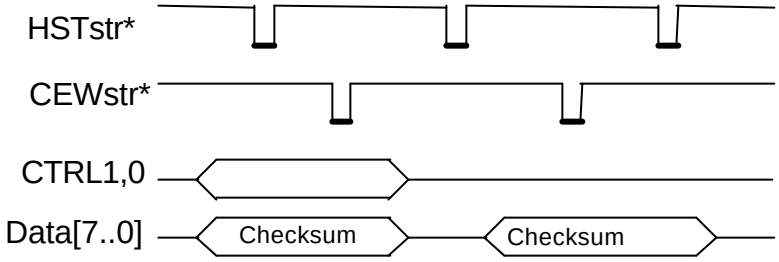


Figure 6 Host to CEWay Checksum Sequence

Timing Diagrams for Parallel Interface

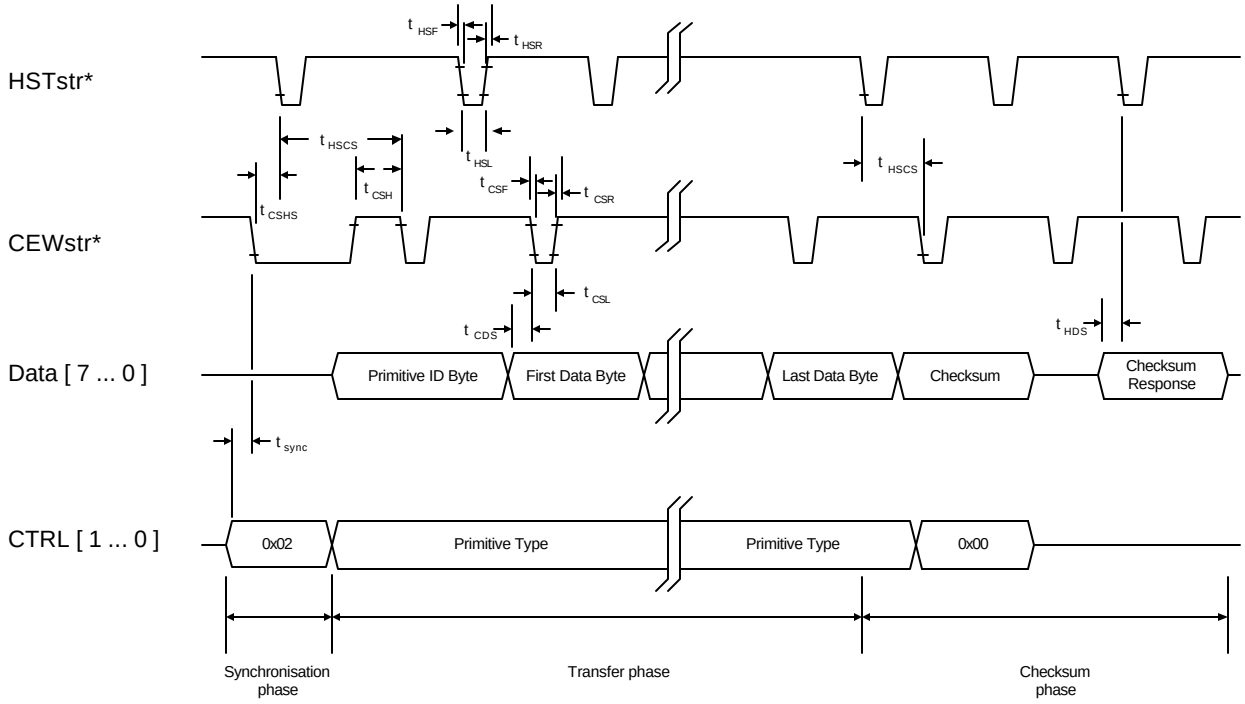


Figure 7 CEWay PL-III to Host Timing Diagram

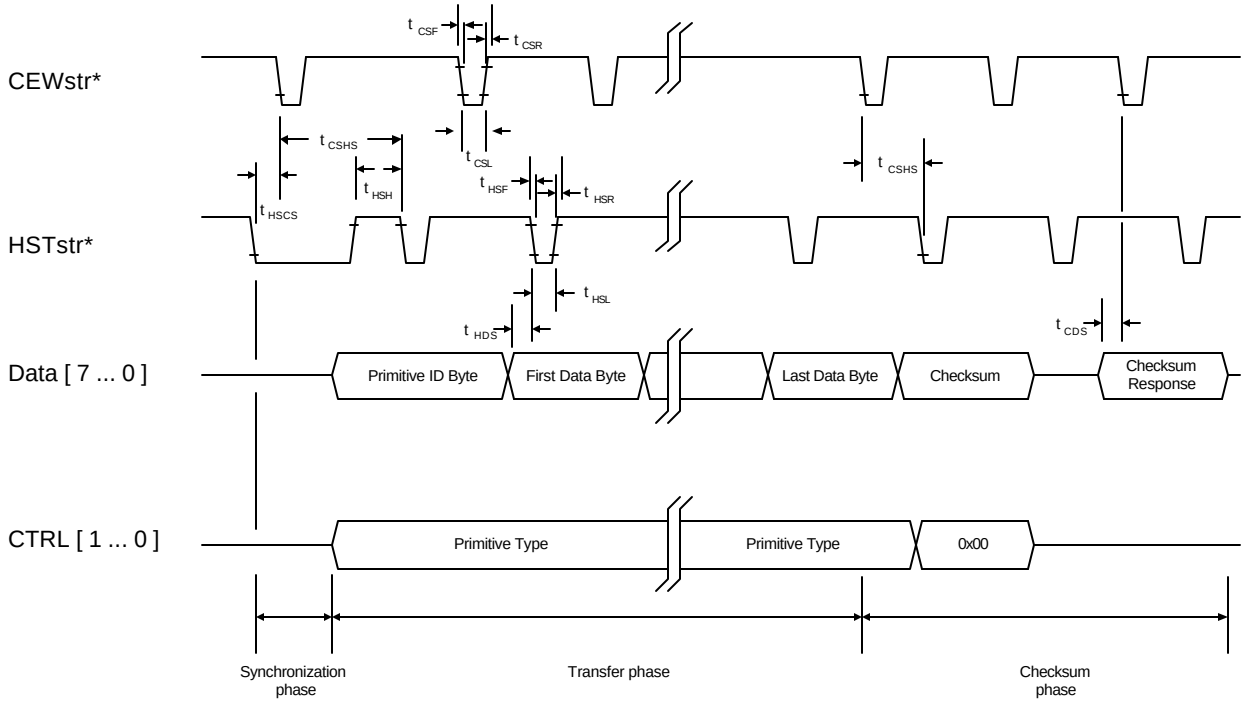


Figure 8 Host to CEWay PL-III Timing Diagram

Table 11 CEWay PL-III Rise and Fall Times for Parallel Interface

Name	Min[1]	Typ[1]	Max[1]	Units	Explanation
t_{CSF}	0.5	0.8	1.2	ns	Fall time for CEWstr*
t_{CSR}	0.5	0.8	1.3	ns	Rise time for CEWstr*
t_{CSL}	0.84	0.84	300[2]	µs	Time that CEWstr* is low
t_{HSF}	0	-	-	ns	Fall time for HSTstr*
t_{HSR}	0	-	-	ns	Rise time for HSTstr *
t_{HSL}	0.84	-	-	µs	Time that HSTstr * is low
t_{sync}	70	70	650	µs	Time that control lines are held to 0x02 before 1st CEWstr* pulse is sent.
t_{CSH}	17.6	17.6	600	µs	Time that CEWstr* is high between synchronization phase and first CEWstr* pulse.
t_{HSH}	0.84	-	-	µs	Time that HSTstr* is high between synchronization phase and first HSTstr* pulse.
t_{CSHS}	0	-	-	µs	Time between CEWstr* negative edge and HSTstr* pulse
t_{HSCS}	17.6	30	75000[3]	µs	Time between HSTstr* negative edge and CEWstr* pulse.
t_{CDS}	5.8	10	300[2]	µs	CEWay data setup time
t_{HDS}	0	-	-	µs	Host data setup time

[1] Rise and fall times assume a 50 pF load at the pin.

[2] This maximum value can only occur when the configuration supporting 27 group addresses is selected. Any configuration that uses 4 or fewer group addresses will have a maximum value of 100 µs.

[3] This maximum value can only occur when sending a 2nd MA_DATA.request [Packet_Request] without having received an MA_DATA.confirm [Packet_Confirm] from the first one (Higher Priority request). Otherwise, the maximum time is 600 µs.

Serial Interface

In this second mode, the CEWay PL-III uses the serial interface to communicate with the host processor. This mode is selected when SELIN is set to 0 (Gnd). The three LSBits of the I/O (BR) port are used to configure the baud rate of the serial interface as described in the table below. The upper five bits are ignored, and should be left floating or tied to V_{DD} to slightly reduce power consumption. It is recommended that the baud rate between the PL-III and the Host be 19,200 baud, as the 9,600 baud rate can result in data overflow if multiple packets are being received. The serial interface parameters are 8 data bits, no parity and one stop bit.

Table 12 Baud Rate Selection for Serial Interface

Value applied on I/O (BR)	*Nominal Baud Rate*	*Actual Baud Rate*	*Discrepancy*
XXXX X101 bin	9,600	9,521	-0.8%
XXXX X110 bin	19,200	19,457	+1.3%

The communication protocol between the CEWay and the host is described below.

Transfer Phase

The transmitter sends the message to transmit with the following format.

<start> <primitive type> <primitive ID byte> <number of data bytes> <data> <checksum>

Start: The start character (0x10 hex) is always used to begin each message.

Primitive type: The primitive type to be sent.

Primitive ID byte: Corresponds to the specific primitive within a primitive type.

Number of data bytes: Number of data bytes included in the data field.

Data: Data field to be sent to the receiver. It should contain the number of bytes indicated by the previous field, <number of data bytes>.

Checksum: This byte is a two's complement of all preceding bytes sent during the transfer phase, excluding the start character.

At the end of the transfer, the receiver must compute the checksum and reply with the frame status as described below.

Table 13 Checksum Response Codes for Serial Interface

Value Sent to the Transmitter	Checksum STATUS
0x00 Hex	Transfer OK
0x01 Hex	Error in Transfer

Serial Backoff

Although the serial interface to the PL-III is full duplex, the chip cannot handle concurrent requests and indications. Therefore, if the host should begin receiving data from the PL-III, be it an indication or a confirm, the host ***must*** stop any request that it might be sending down to the PL-III, and enter into receive mode. This higher priority in the interface is conferred upon the PL-III to minimize risks of losing data in reception.

The PL-III will not deliberately start sending up data once it realizes that the host is descending a request. However, due to the nature of the serial interface, it is possible that the host receives the first byte of an indication or confirm anywhere up to the end of the transmission of the third byte (the Primitive ID byte) of its request. Once the fourth byte (Number of Data Bytes) of a request has commenced transmission, the host can be sure that the PL-III will not attempt to interrupt it.

CEWay Primitives

The CEWay primitives allow communication between the PL-III and the Host processor. The communication is performed via the Host Interface (parallel or serial). The CEWay has both input and output primitives.

CEWay Input Primitives

There are two types of CEWay input primitives: Data Requests and LSM Requests.

Data Requests: Primitive type = 0x01

MA_DATA.request;
MA_ACK_DATA.request.

LSM Requests: Primitive type = 0x02

MA_SET_VALUE.request;
MA_READ_VALUE.request;
LSM_EVENT.indication;
MA_INITIALIZE_PROTOCOL.request.

Data Requests

To use any of these primitives, the **Primitive Type** must be set to **0x01**.

MA_DATA.request or **MA_ACK_DATA.request** (MA_ACK_DATA.request is performed with the *Ack_Service* parameter, within *DLL_Service_Level*, being set.) Ref: EIA 600.41, sections 1.1.3.2.1 and 1.1.4.2.1			
Primitive ID Byte	**Primitive ID Name**	**Description**	
0x00	**Packet_ Request**	This primitive is used to request that a data packet be transmitted on the medium.	
Byte #	*Field Name*		
1	*DLL_Service_ Level*		
	Bits	*Parameter*	
	0	*Addressed_Service*. When set, use addressed service (ADR_UNACK_DATA or ADR_ACK_DATA), otherwise use non-addressed service (UNACK_DATA or ACK_DATA).	
	1	*Ack_Service*. When set, use Acknowledged service (ACK_DATA or ADR_ACK_DATA), otherwise use Unacknowledged service (UNACK_DATA or ADR_UNACK_DATA).	
	2	*Reserved* (configurable via *Prohibit_Illegal_Services*)	
	4-3	*Priority*. Use the following priority service.	
		Priority values	Description
		00	HIGH
		01	STANDARD
		10	DEFERRED
		11	Reserved (configurable via *Prohibit_Illegal_Services*)
	5	*Include_Source*. When set, includes the source addresses, otherwise does not include the source addresses if *Addressed_Service* is not set. The source is always included when *Addressed_Service* is set so this parameter becomes N/A.	
	6	*Extended_Service_Class*. When set, uses EXTENDED Service Class, otherwise uses BASIC Service Class.	
	7	*Skip_DHC*. When set, do not pass the *Destination House Code (DHC)* down, since MACS will use the same one as its local Source House Code.	
2, 3	*Destination Address (DA)*	Least significant byte first.	

If Skip_DHC = 0 *4, 5*	*Destination House Code (DHC)*	Least significant byte first. These bytes only apply if the *Skip_DHC* byte in the *DLL_Service_Level* field was set to 0, otherwise no house code should be passed down, and the packet will be transmitted to the same house code as its Source House Code (SHC).
If Skip_DHC = 1 *[4 – 35]* *If Skip_DHC = 0* *[6 – 37]*	*Data*	Most significant byte first. Minimum length is 0 bytes, maximum length is 32 bytes.

Primitive ID Byte	**Primitive ID Name**	**Description**
0x10	**Abort_ Request**	This primitive is used to abort the previous MA_DATA.request [Packet_Request] that is currently in progress, for which an MA_DATA.confirm [Packet_Confirm] has not been returned yet. An MA_DATA.confirm [Packet_Confirm] will be returned to indicate whether the MA_DATA.request [Packet_Request] was successful or not. If no data request is in progress, nothing will be returned.

LSM Requests

To use any of these primitives, the **Primitive Type** must be set to **0x02**.

MA_SET_VALUE.request
Ref: EIA 600.42 section 1.2.5.2.3

Primitive ID Byte	**Primitive ID Name**	**Description**		
0x00	**Configuration**	This primitive is used to configure the CEWay. It may only be (and must be) called after receiving an MA_INITIALIZE_PROTOCOL.confirm [SUCCESS] generated by a hardware reset. It contains 13 bytes + optional group addresses.		
Byte #	*Field Name*			
1	*CEWay_Configuration_Reg1*			
	Bits	*Description*		
	1 – 0	*CEWay_Configuration_Mode*. Use CEWay in the following configuration.		
		Mode Value	Description	Suggested values
		00	Normal DLL	00
		01	Reserved	
		10	Reserved	
		11	Reserved.	
	2	*Prohibit_Illegal_Services*. (See appendix A)		1
	3	*Prohibit_Extended_Services*. (See appendix A)		1
	4	*Non_Adr_Ack_Data_Result*. When set, IACK or FAILURE packets are returned with a specific result in the Data field (ref: EIA 600.42, Appendix A). Otherwise, they are sent with a Null field Leading Zero Suppressed.		1
	5	*IACK_IRETRY_PRE_8_SYM*. When set, IACK, FAILURE, ADR_IACK and Immediate Retry (IRetry) packets are sent with a full Preamble field containing 8 Preamble symbols and a PEOF. Otherwise, these packets are sent without Preamble symbols but only a PEOF (ref: EIA 600.41 sections 3.1.1 and 3.1.2).		0
	6	*NUM_RETRIES*. When set, if Acknowledged service is used, this is used to send 1 IRetry. Otherwise, no IRetry is sent (ref: EIA 600.41, section 3.2.4.3).		0
	7	*Reserved*. Must be set to 0.		0

2	*CEWay_Configuration_Reg2*		
	Bits	*Description*	Suggested values
	0	*Automatic_Upper_Layer_Busy*. If this bit is set, the CEWay will automatically set itself UPPER_LAYER_BUSY after receiving any good data packet that is addressed to it. Once the packet has been passed up to the host, and the host is ready to receive another packet, the host must pass the CEWay an LSM_EVENT.indication [UPPER_LAYER_NOT_BUSY] to clear this condition (ref: EIA 600.42, section 1.2.4.2.7.2).	0
	1	*Wait_While_Tx_Busy_List*. When set, if an Addressed frame is passed to the CEWay for transmission, but its Tx_Dup_List is full, it will wait until an element frees itself so that it can transmit the new packet. If the Retrans_Time timer expires before this occurs, then an MA_DATA.confirm [Packet_Confirm, FAILURE_MACS_TX_LOCAL_BUSY_EXPIRED_LIST] will be passed to the host. If this bit is not set, and an Addressed frame is passed to the CEWay for transmission when its Tx_Dup_List is full, then an MA_DATA.confirm [Packet_Confirm, FAILURE_MACS_TX_LOCAL_BUSY_LIST] will immediately be passed back to the host, and the packet will be discarded.	1
	2	*Support_Statistical_Counters_Overflow*. If this bit is set, an MA_DATA.indication [STATISTICAL_COUNTER_OVERFLOW] will be passed to the host whenever one of the 8 statistical counters overflows. Otherwise, nothing is done when a statistical counter overflows.	0
	3	*Support_Diagnostic_Indication*. When set, enables MA_EVENT.indication [RX_REFUSED] to be passed from MACS LSM to the host. When set, enables MA_EVENT.indication [RX_OVERRUN] to be passed from MACS LSM to the Host. If this bit is not set, nothing is done when a packet is lost in reception.	0

	4	*Support_Busy_After_Reset*. When set, the Rx and Tx Dup_Lists are both set busy for their full duration following a chip reset. This is to prevent received duplicate packets being interpreted as new ones, and new transmitted packets being interpreted as duplicates by the receiving node. No packets will be passed up to the host for the full duration of the Rx_Dup_List (0.9375 seconds).	0
	6 – 5	*Reserved*. Must be set to 0.	0
	7	*Support_Home Plug & Play*. When set, packets addressed to the Home Plug & Play broadcast address (DA = 0xFFFF) are passed up to the host. Otherwise they are discarded.	1

3	*CEWay_Configuration_Reg3*						
	bits	*Description*					
	3 – 0	*Memory_Configuration*. These four bits are used to specify how memory will be allocated between four items: group addresses, the Tx_Dup_List, the Rx_Dup_List, and Statistical_Counters. The sixteen possible configurations are listed below (ref: EIA 600.41, section 3.2.3.2. EIA 600.42, sections 3.5.26 and 3.5.26).					
		Mode value	Group Addresses	Tx_Dup_List elements	Rx_Dup_List elements	Statistical_Counters	Sugges-ted Value
		0000	3	4	4	Yes	1111
		0001	0	9	1	Yes	
		0010	2	1	7	Yes	
		0011	20	1	1	Yes	
		0100	2	0	8	Yes	
		0101	15	2	2	Yes	
		0110	9	2	4	Yes	
		0111	3	3	5	Yes	
		1000	3	6	4	No	
		1001	2	10	1	No	
		1010	3	1	8	No	
		1011	0	0	10	No	
		1100	27	0	1	No	
		1101	13	2	4	No	
		1110	7	2	6	No	
		1111	4	3	6	No	
	7 - 4	*Reserved*. Must be set to 0.					0000
4	*Max_Restart*	Range from 0 to 250					250

5	*Unack_Ch_Access_Num*	Range from 0 to 250	4
6	*Ch_Access_Period.*	Contains both Unack and Ack Channel Access Periods.	0
	Bits	*Description*	
	3 – 0	*Unack_Ch_Access_Period* (x25 ms). Range from 0 to 15 (375 ms).	0
	7 - 4	*Ack_Ch_Access_Period* (x25 ms). Range from 0 to 15 (375 ms).	0
7	*Unack_Retrans_Time*	Units of 25 ms. Range from 0 to 30 (750 ms).	30 (750 ms)
8	*Reserved.*		0
9	*Ack_Retrans_Time*	Units of 25 ms. Range from 0 to 30 (750 ms).	30 (750 ms)
10, 11	*Unit Address1 (UA1)*	***2 bytes, least significant byte first***. Range from 0 to 0xFFFF. Note that only addresses from 0x0001 – 0x00FD, 0x1001 – 0x7FFF, and 0x8001 – 0xEFFF are valid unit addresses within the CEBus standard (ref EIA 600.42, section 2.5.4).	N/A
12, 13	*House Code1 (HC1)*	***2 bytes, least significant byte first***. Range from 0 to 0xFFFF. Note that only addresses from 0x0001 – 0x00FF, 0x0101 – 0x7FFF, and 0x8001 – 0xEFFF are valid house codes within the CEBus standard (ref EIA 600.42, section 2.5.5).	N/A
[14 – 67]	*Group Addresses*	Between 0 and 27 group addresses can follow the house code, depending on what memory configuration was selected. Unused group addresses for a given memory configuration need not be passed down, and will not be used by the MACS. For example, if memory configuration 15 is selected, but only 2 of the available 4 group addresses are currently used, the two unused group addresses do not need to be passed down. Each group address is ***2 bytes, least significant byte first***. They should range from 0x0101 to 0x0FFF (ref EIA 600.42, section 2.5.4).	N/A

MA_READ_VALUE.request
Ref: EIA 600.42, section 1.2.5.2.1

Primitive ID Byte	Primitive ID Name	Description
0x20	**Configuration**	This primitive is used to read the configuration information that was passed down during the MA_SET_VALUE.request [Configuration]. The CEWay will respond with an MA_READ_VALUE.confirm [Configuration] containing the 13 bytes of configuration data, as well as the values of all group addresses supported in the current configuration.
0x24	**Statistical_ Counters**	This primitive is used to read the CEWay's statistical counters (see MA_READ_VALUE.confirm [Statistical_ Counters] for descriptions.) It can only be called when there are no outstanding MA_DATA.requests [Packet_Request] to the CEWay, ie. the host is not waiting for any MA_DATA.confirms [Packet_Confirm]. The CEWay will respond with an MA_READ_VALUE.confirm [Statistical_Counters] containing 8 bytes of data. Note that this data is only valid if a Memory Configuration using statistical counters was selected (via *CEWay_Configuration_Reg3*).

LSM_EVENT.indication
This primitive allows the LSM to indicate to MACS and PLSES that a major event has happened.
Ref: EIA 600.42, section 1.2.5.2.7.2

Primitive ID Byte	Primitive ID Name	Description
0x80	**UPPER_ LAYER_NOT_ BUSY**	Used by the host to indicate to the CEWay that it is now ready to receive new packets. Only needs to be used after previously using the LSM_EVENT.indication [UPPER_LAYER_BUSY] primitive, or after receiving a packet with the *Automatic_Upper_Layer_Busy* parameter set.
0x81	**UPPER_ LAYER_BUSY**	Used by the host to indicate that it is too busy to accept any further packets from the CEWay. Once the CEWay is UPPER_LAYER_BUSY, it will return MA_DATA.confirms [Packet_Confirm, FAILURE_REMOTE_BUSY (0x0001)] for all Acknowledged packets that it receives, while simply discarding Unacknowledged packets. The host needs to use the LSM_EVENT.indication [UPPER_LAYER_NOT_BUSY] primitive when it is ready to receive new packets.

MA_INITIALIZE_PROTOCOL.request
This primitive allows the LSM to reset MACS and PLSES.
Ref: EIA 600.42, section 1.2.5.2.1

Primitive ID Byte	Primitive ID Name	Description
0xA0	**RESET**	Performs a software reset on the PL-III. An MA_INITIALIZE_PROTOCOL.confirm [SUCCESS] should be returned within 50 ms to indicate that the node was successfully reset.

CEWay Output Primitives

There are three types of CEWay output primitives: the Data Indications, the Request Confirms and the LSM Indications.

Data Indications: Primitive type = 0x01

MA_DATA.indication;
MA_ACK_DATA.indication.

Request Confirms: Primitive type = 0x02

MA_DATA.confirm;
MA_ACK_DATA.confirm;
MA_READ_VALUE.confirm;
MA_INITIALIZE_PROTOCOL.confirm.

LSM Indications: Primitive type = 0x03

MA_EVENT.indication;
MA_FAILURE_REPORT.indication;
PH_EVENT.indication.

Data Indications

When the CEWay PL-III uses any of these primitives, it will set the **Primitive Type** to **0x01**.

MA_DATA.indication or **MA_ACK_DATA.indication** (MA_ACK_DATA.indication is performed via an MA_DATA.indication, with the *Ack_Service* parameter, within Indication Parameters, being set). Ref: EIA 600.42, sections 1.1.3.2.2 and 1.1.4.2.2			
Primitive ID Byte	**Primitive ID Name**	**Description**	
0x00	**Packet_ Indication**	This primitive allows the MACS to indicate to the HOST that it has received a frame and to pass it up to the HOST.	
Byte #	*Field Name*		
1	*Indication_ Parameters*		
	Bits	*Parameter*	
	0	*Addressed_Service*. When set, indicates that an addressed service was used in the received packet. Otherwise, it means that a non-addressed service was used.	
	1	*Ack_Service*. When set, indicates that an Acknowledged service was used in the received packet. Otherwise, it means an Unacknowledged service was used.	
	2	*Rx_Ch_Noisy*. When set, indicates that the PL-III had difficulty receiving the packet, perhaps indicating a weak signal or a noisy channel. Otherwise, the packet was received without difficulty.	
	4-3	*Priority*. Indicates the following priority was used in the received packet.	
		Priority values	Description
		00	HIGH
		01	STANDARD
		10	DEFERRED
		11	Reserved (configurable via *Prohibit_Illegal_Services*)
	5	*Rx_Include_Source*. When set, indicates that the received packet included its source addresses. Otherwise it did not include its source addresses.	
	6	*Rx_Extended_Service_Class*. When set, indicates that the received packet used EXTENDED Service Class. Otherwise, it used BASIC Service Class.	
	7	Reserved. Will be set to 0.	
2, 3	*Destination Address (DA)*	Least significant byte first.	

4, 5	*Destination House Code (DHC)*	Least significant byte first.
6, 7	*Source Address (SA)*	Least significant byte first.
8, 9	*Source House Code (SHC)*	Least significant byte first.
10 - 41	*Data*	Most significant byte first. Minimum length is 0 byte, maximum length is 32 bytes.

Request Confirms

When CEWay PLIII uses any of these primitives, it will set the **Primitive Type** to **0x02**.

MA_DATA.confirm or **MA_ACK_DATA.confirm** (Same as MA_DATA.confirm, with extra *Confirm_Results*). Ref: EIA 600.42, sections 1.1.3.2.3 and 1.1.4.2.3		
Primitive ID Byte	**Primitive ID Name**	**Description**
0x00	**Packet_ Confirm**	This primitive allows the MACS to confirm to the HOST that it has attempted to transmit the request and to pass the *Confirm_Result* up to the HOST. See Appendix B for more detailed information on the possible confirm codes.
Byte #	*Field Name*	
1, 2	*Confirm_Result*	Most significant byte first.
	Hex	
	0000	REMOTE_SUCCESS
	0001 – 007F	REMOTE_BUSY RANGE
	0010	REMOTE_BUSY_RCV_LIST
	0020	REMOTE_BUSY_LIST_RESET
	0080 – 00FF	REMOTE_REJECT RANGE
	0090	REMOTE_REJECT_EXTENDED
	0100 – DFFF	REMOTE_OTHER RANGE
	E000	REMOTE_OTHER_OUT_OF_RANGE
	F000	LOCAL_SUCCESS
	F110 – F11F	FAILURE_NO_ACKNOWLEDGE RANGE
	F110	FAILURE_NO_ACKNOWLEDGE
	F111	FAILURE_NO_ACKNOWLEDGE_BAD_FRAME
	F112	FAILURE_NO_ACKNOWLEDGE_RETRY_BACKOFF
	F113	FAILURE_NO_ACKNOWLEDGE_WRONG_TYPE
	F114	FAILURE_NO_ACKNOWLEDGE_EXPIRED
	F115	FAILURE_NO_ACKNOWLEDGE_ADR_IACK
	F116	FAILURE_NO_ACKNOWLEDGE_HIGH_PRIORITY
	F120 – F12F	FAILURE_EXCESSIVE_COLLISIONS RANGE
	F120	FAILURE_EXCESSIVE_COLLISIONS
	F123	FAILURE_EXCESSIVE_COLLISIONS_EXPIRED
	F124	FAILURE_EXCESSIVE_COLLISIONS_ACCESS_NUM
	F130 – F13F	FAILURE_MACS_TX_LOCAL_BUSY RANGE
	F131	FAILURE_MACS_TX_LOCAL_BUSY_LIST
	F132	FAILURE_MACS_TX_LOCAL_BUSY_EXPIRED_LIST
	F133	FAILURE_MACS_TX_LOCAL_BUSY_RESET
	F140 – F14F	FAILURE_MACS_HIGHER_PRIORITY RANGE
	F140	FAILURE_MACS_HIGHER_PRIORITY
	F150 – F15F	FAILURE_MACS_ILLEGAL_SERVICES RANGE

	F150	FAILURE_MACS_ILLEGAL_SERVICES
	F151	FAILURE_MACS_ILLEGAL_SERVICES_PRIORITY
	F152	FAILURE_MACS_ILLEGAL_SERVICES_XMIT_ADDRESS

MA_READ_VALUE.confirm
Ref: EIA 600.42, section 1.2.5.2.6

Primitive ID Byte	**Primitive ID Name**	**Description**		
0x20	**Configuration**	A MA_READ_VALUE.request [Configuration] is responded to with this confirm, which contains the data passed down during the initial MA_SET_VALUE.request [Configuration], as well as the Device ID. The values for all group addresses that are supported in the current configuration will be passed up, even those that were never initialized, which will have values of 0x0000.		
Byte #	*Field Name*			
1	*Device ID*	This byte represents what type and version of chip is being used.		
	Bits	*Description*		
	7 –4	*Device_Type*. This upper nibble represents the type of device		
		Value	Device Name	
		0000	PL-III	
	3 - 0		*Device_Version*. This lower nibble represents the device version.	
			Value	Version Name
			0000	Version 1.3
2	*CEWay_Configuration_Reg1*			
3	*CEWay_Configuration_Reg2*			
4	*CEWay_Configuration_Reg3*			
5	*Max_Restart*			
6	*Unack_Ch_Access_Num*			
7	*Ch_Access_Period*			
8	*Unack_Retrans_Time*			
9	*Reserved*			

10	*Ack_Retrans_ Time*	
11, 12	*Unit Address*	
13, 14	*House Code*	
[15 – 68]	*Group Addresses*	

Primitive ID Byte	**Primitive ID Name**	**Description**
0x24	**Statistical_ Counters**	A MA_READ_VALUE.request [Statistical_Counters] is responded to with this confirm. It contains 8 bytes of data, which are only valid if a memory configuration supporting statistical counters was selected via *CEWay_Configuration_Reg3*. Note that the statistical counters can only be read when there are no outstanding MA_DATA.requests [Packet_Request] to the CEWay.
Byte #	*Field Name*	
1	*Rx_Carrier_ Lost_Count*	This counter contains the number of detected “Carrier Lost “ in reception since it was last read.
2	*Rx_Good_ Frame_Count*	This counter contains the number of valid frames received (including Acknowledgements) since it was last read.
3	*Rx_Crc_Error_ Frame_Count*	This counter contains the number of frames received with CRC errors since it was last read.
4	*Rx_Error_ Frame_Count*	This counter contains the number of frames received with errors, including CRC errors, since it was last read.
5	*Tx_Good_ Request_Count*	This counter contains the number of requests successfully transmitted since it was last read. A request that received any success result for its MA_DATA.confirm [Packet_Confirm] is considered as one successfully transmitted request.
6	*Tx_Failed_ Request_Count*	This counter contains the number of requests unsuccessfully transmitted since it was last read. A request that received any failure result for its MA_DATA.confirm [Packet_Confirm] is considered as one unsuccessfully transmitted request.
7	*Tx_Ack_ Retrans_Count*	This counter contains the number of frames successfully transmitted, using Acknowledged services (including retransmissions and IRetries) since it was last read.
8	*Tx_Collision_ Count*	This counter contains the number of frames transmitted that encountered a collision since it was last read.

MA_INITIALIZE_PROTOCOL.confirm Ref: EIA 600.42, section 1.2.5.2.2		
Primitive ID Byte	**Primitive ID Name**	**Description**
0xC0	**SUCCESS**	This primitive allows the MACS to indicate to the HOST that it has been successfully reset by the host (via the RESET* pin or by an MA_INITIALIZE_PROTOCOL.request), or that an interface error has occurred and the MACS has reset itself. If the reset originated from the host, then it must perform an MA_SET_VALUE.request to finish initializing the MACS. If the reset was due to an interface error, it is recommended that the host resets the CEWay itself before continuing with the MA_SET_VALUE.request.
N/A	**FAILURE**	When no MA_INITIALIZE_PROTOCOL.confirm [SUCCESS] is sent within 50 ms following a hardware or software reset.

LSM Indications

When CEWay PL-III uses any of these primitives, it will set the **Primitive Type** to **0x03**.

MA_EVENT.indication This primitive allows the MACS to indicate to the HOST that a major event has occurred. (Additional information not covered by EIA 600.42 specification.)		
Primitive ID Byte	**Primitive ID Name**	**Description**
0x02	**STATISTICAL_ COUNTER_ OVERFLOW**	One or more of the statistical counters has overflowed from 255 to 0. The following byte of data contains 8 flags to indicate which counters have just overflowed. The MACS must be configured to use statistical counters (via CEWay_Configuartion_Reg3) for this primitive to occur.
Byte #	*Field Name*	
1	*Overflow Flags*	
	Bits	*Description*
	0	*Rx_Carrier_Lost_Count*
	1	*Rx_Good_Frame_Count*
	2	*Rx_Crc_Error_Frame_Count*
	3	*Rx_Error_Frame_Count*
	4	*Tx_Good_Request_Count*
	5	*Tx_Failed_Request_Count*
	6	*Tx_Ack_Retrans_Count*
	7	*Tx_Collision_Count*
Primitive ID Byte	**Primitive ID Name**	**Description**
0x03	**RX_ OVERRUN**	Will be set if a packet (not necessarily addressed to the node) is lost for reasons other than the ones described in RX_REFUSED. This primitive is configurable via *CEWay_Configuration_Reg2*, *Support_Diagnostic_Indication*.
0x04	**RX_ REFUSED**	Will be set if a packet addressed to the node is lost in reception. This can happen if the node is either UPPER_LAYER_BUSY, BUSY_AFTER_RESET, or if the Rx_Dup_List is full. This primitive is configurable via *CEWay_Configuration_Reg2*, *Support_Diagnostic_Indication*.

MA_FAILURE_REPORT.indication
This primitive allows the MACS to indicate a software or hardware error.
Ref: EIA 600.42 section 1.2.5.2.8

Primitive ID Byte	Primitive Name	Description
0x20	**GENERAL_ FAILURE**	Indicates that one or more failures have occurred in the Physical and/or Data Link Layers. The exact nature of the failure(s) is contained within this primitive's data byte. The chip should always be reset whenever this primitive is received.
Byte #	*Field Name*	
1	*Failure Description*	
	Bits	
	4 – 0	*MAC_FAILURE* (MACS firmware error). These five bits are used to indicate internal errors in the MACS firmware.
	5	*PH_FAILURE* (No_CH_QUIET_Sent). The MACS firmware sometimes expects PH_CC_STATUS.indication (CH_QUIET) from the Physical layer. If these indications are not received during a 25 ms period, this bit will be set.
	6	*PH_FAILURE* (No_PH_Confirm_Sent). The MACS firmware expects a PH_CC_DATA.confirm after every symbol that is transmitted. If this confirm is not received within 25 ms, this bit will be set.
	7	Reserved

PH_EVENT.indication
This primitive allows the PLSES to indicate to the HOST that a major event has occurred.
Ref: IS-60.03, section 4.2.5

Primitive ID Byte	Primitive Name	Description
0x40	**PH_TX_ JABBER_ DETECT**	This condition occurs when 1000 consecutive superior symbols have been transmitted by the node. When the HOST receives a PH_EVENT.indication [PH_TX_JABBER_DETECT], the host needs to immediately perform a hardware reset on the PL-III and then wait 10 000 USTs (1 sec) before reinitializing CEWay by performing an MA_SET_VALUE.request. Ref EIA 600.31, section 5.7.2
0x41	**PH_RX_ JABBER_ DETECT**	This condition occurs when 1000 consecutive superior symbols have been detected in reception. The host needs to perform a hardware reset on the PL-III and then reinitialize the CEWay by performing an MA_SET_VALUE.request.

Proper Usage of the Primitives

Although the previous section detailed the syntactical use of each primitive, it did not offer much help as to when these primitives should or can be used. Here are some guidelines about how the various primitives are best used.

PL-III Initialization

Reset PL-III

The preferred method to reset the PL-III is via its dedicated RESET* line (pin 32). This pin must be held low for at least 1.68 µs to perform a complete reset of the chip. Another alternative is to use the MA_INITIALIZE_PROTOCOL.request primitive. Note that the software reset may be difficult to use if an interface error has occurred.

Wait for MA_INITIALIZE_PROTOCOL.confirm [SUCCESS]

Once reset, the MACS will attempt to indicate this condition to the host via the MA_INITIALIZE_PROTOCOL.confirm [SUCCESS] primitive. This primitive should start being received within 50 ms of a reset condition. The time it takes to fully receive the primitive is heavily dependent on the host in the case of the parallel interface, or on the selected baud rate in the case of the serial interface.

Send MA_SET_VALUE.request [Configuration]

Once an MA_INITIALIZE_PROTOCOL.confirm [SUCCESS] has been received, the host must initialize the MACS firmware's various parameters via an MA_SET_VALUE.request [Configuration]. **Note that this is the only time that this primitive may be called; it is not possible to change the MACS parameters "on the fly", as this could cause problems if any address should change. If it is wished to change some parameters, the chip must first be reset, and an MA_INITIALIZE_PROTOCOL.confirm [SUCCESS] be received, before another MA_SET_VALUE.request [Configuration] may be performed.**

There are many parameters that can be specified within the MA_SET_VALUE.request. Those that are not described in the primitive itself are listed and explained below. Once the MA_SET_VALUE.request is finished, the chip is initialized and ready to receive or transmit packets.

Max_Restart

This parameter specifies the maximum number of times that channel access will be attempted when transmitting a packet using a Non-Addressed service. Unless it is wished to limit the number of attempts made to access the channel, it is recommended that this parameter be set to 250. An MA_DATA.confirm [Packet_Confirm, FAILURE_EXCESSIVE_COLLISIONS (0xF120)] will be passed up should the packet not be transmitted within *Max_Restart* channel access attempts. Note that if this parameter is set to 0, the packet will only be transmitted if no other node preempts the MACS from transmitting during its first channel access. If the MACS is already in the middle of a reception when it receives the MA_DATA.request [Packet_Request], its first attempt at channel access will only occur once the channel is Quiet.

(Ref: EIA 600.41, sections 3.2.2.2, 3.2.3.1, and 3.2.4.3)

Unack_Ch_Access_Num

This parameter specifies the number of times all ADR_UNACK_DATA packets will be transmitted. For example, if this parameter were set to 4, then the packet would be transmitted once, and retransmitted three times. The only exception to this is if this number of transmissions was not completed within the Retrans_Time (the smaller of MAX_RETRANS_TIME, which is 750 ms, and *Unack_Retrans_Time*). If the packet was transmitted at least once, an MA_DATA.confirm [Packet_Confirm, LOCAL_SUCCESS (0xF000)] will be passed up, otherwise an MA_DATA.confirm [Packet_Confirm, FAILURE_EXCESSIVE_COLLISIONS_EXPIRED (0xF123)] will be returned. Note that if this parameter is set to 0 and an Unacknowledged packet is passed down for transmission, an MA_DATA.confirm [Packet_Confirm, FAILURE_EXCESSIVE_COLLISIONS_ACCESS_NUM (0xF124)] will be returned.

(Ref: EIA 600.41, sections 3.2.3.1 and 3.4)

Ch_Access_Period

This variable contains two separate parameters, *Unack_Ch_Access_Period* in the lower nibble and *Ack_Ch_Access_Period* in the upper nibble. Both parameters are denominated in units of 25 ms, and can range from 0 to 15 (0 ms to 375 ms). The purpose of these two parameters is to control how quickly retransmissions of Unacknowledged and Acknowledged packets take place. When a packet has been transmitted once, the next retransmission (excluding IRetries), will not be attempted until a time equivalent to the appropriate *Ch_Access_Period* parameter has expired. This retransmission, if unsuccessful, will continue to be re-attempted using the normal channel access delays until it is successful, or until some other condition causes the retransmission to be aborted. Note that if this parameter is set to 0, the MACS will still respect its regular channel access delay before attempting to retransmit.

Unack_Retrans_Time

This parameter is used to specify the maximum period of time (in units of 25 ms) during which channel access attempts (transmissions and retransmissions) will be attempted for Unacknowledged packets. An Unacknowledged packet transmission will finish, and an MA_DATA.confirm [Packet_Confirm] will be passed up, if this timer should expire before the packet was transmitted *Unack_Ch_Access_Num* of times. The *Confirm_Result* will either be LOCAL_SUCCESS (0xF000) if the packet was transmitted at least once, or FAILURE_EXCESSIVE_COLLISIONS_EXPIRED (0xF123), if the packet was never transmitted. Note that if this parameter is set to 0, the packet will only be transmitted if no other node preempts the MACS from transmitting at its first channel access and that no packet is already being received when the MACS gets ready to transmit its own.

(Ref: EIA 600.41, sections 3.2.3.1, 3.2.3.2.1, and 3.2.6)

Ack_Retrans_Time

This parameter is the Acknowledged equivalent of the *Unack_Retrans_Time*, and is specified in units of 25 ms. It returns the same MA_DATA.confirm [Packet_Confirm, FAILURE_EXCESSIVE_COLLISIONS_EXPIRED (0xF123)] if the packet is not transmitted within *Ack_Retrans_Time*. If the packet is transmitted and then the timer expires, MACS will always wait the usual IACK Wait_Time for the IACK before actually returning any MA_DATA.confirm [Packet_Result].

(Ref: EIA 600.41, sections 3.2.3.2.1, 3.2.5, and 3.2.6)

Transmitting a Packet

There is only one way to transmit a packet using the PL-III, and that is via the MA_DATA.request or MA_ACK_DATA.request, which are both mapped to the same primitive. This combined primitive will simply be referred to as an MA_DATA.request [Packet_Request].

DLL_Service_Level

The first byte to be passed down (other than the interface protocol bytes dependent on whether a parallel or serial interface is used), is the *DLL_Service_Level* byte. The first two bits within this byte are used to specify which of the four available DLL services will be used to transmit the packet: ADR_ACK_DATA, ADR_UNACK_DATA, ACK_DATA, or UNACK_DATA.

Transmit Duplicate List

Note that the use of an addressed service requires that an element be free in the Transmit Duplicate List (Tx_Dup_List). This is a list maintained within the MACS used to signal new or duplicate packets to the receiving node. It is configurable via the lower nibble of *CEWay_Configuration_Reg3*.

Every time an addressed packet is sent to an address not currently in the list, a new element in the Tx_Dup_List is used to maintain an association of a Sequence_Number bit with this new address. This association is maintained during a period of time equal to 1.5 * MAX_RETRANS_TIME (=1.125 seconds), and then that element is freed up.

If an addressed packet is sent to an address already in the list, then the element in the list is re-used, the 1.125 second timer is reset and the Sequence_Number bit is toggled.

If an addressed packet is sent to a new address, but no elements are free within the Tx_Dup_List, then an MA_DATA.confirm [Packet_Confirm, FAILURE_MACS_TX_LOCAL_BUSY_LIST] will be passed up.

(Ref: EIA 600.41, section 3.2.3.2. EIA 600.42, sections 3.5.26 and 3.5.27)

Reserved Bit

This bit within the *DLL_Service_Level* byte should always be set to 0. Its value is of no present importance. However, for test purposes, if the *Prohibit_Illegal_Services* bit is not set during

the MA_SET_VALUE.request, the *Reserved* bit will be copied to the Reserved bit of the Control Field of the packet to be transmitted.

Priority Bits

The next two bits within the *DLL_Service_Level* byte are used to specify the priority of the packet being transmitted. This priority is used to specify the delay that should be added to the base delay for the transmitted packet.

- '00' corresponds to a high priority packet, and should only be used for interactive applications where the response time to user input is critical.
- '01' corresponds to a standard priority packet, which should be used for most inter-node communication. An additional 4-UST delay is imposed on each packet transmission or re-transmission.
- '10' corresponds to a deferred priority packet, which should be used for any background traffic, such as data logging. An additional 8-UST delay is imposed on each packet transmission or re-transmission.
- '11' is an illegal priority value. However, if the *Prohibit_Illegal_Services* bit is not set , the packet will be transmitted with the priority bits in the control field set to this value, with the packet delay corresponding to a deferred packet.

(Ref: EIA 600.41, section 3.2.1.2)

Include_Source Bit

This bit is used to specify whether the source address should be included in the packet to be transmitted. It is only applicable for packets not using an addressed service, since addressed packets will always include the source address.

Extended_Service_Class Bit

This bit maps to the Service_Class bit within the Control Field of the packet to be transmitted. Setting it indicates that the packet uses an Extended service (currently not defined within the CEBus standard), while clearing it indicates that the packet uses the Basic service. It is not recommended to set this bit. Note that the *Prohibit_Extended_Service* bit within *CEWay_Configuration_Reg1* of the MA_SET_VALUE.request only applies to packets received using an Extended service, and has no effect on the operation of this bit in transmission. (Ref: EIA 600.41, section 3.1.1)

Skip_DHC

This bit can be used to save time in the interface by not having to pass down the *Destination House Code (DHC)* for packets being transmitted to the same House Code. If set, only the *Destination Address (DA)* needs to be passed down before the packet data bytes, and the MACS will copy the Source House Code (SHC) into the DHC field of the transmitted packet. If not set, then both the DHC and DA need to be passed down before the packet data bytes.

Destination Node

The next bytes are used to specify which node is going to receive the packet. The first two bytes are used to specify the *Destination Address (DA)*, and must be passed down LSByte first (the order in which the bytes will actually be transmitted.) The DA will be followed by the

DHC (two bytes, LSByte first) if the *Skip_DHC* bit was not set. There is no need to specify either the Source Address (SA) or Source House Code (SHC), since both are stored internally into the MACS following an MA_SET_VALUE.request. Whether the SA and SHC are included within the transmitted packet depends on the *Include_Source* bit or whether an addressed service was used. Note that if the SHC is the same as the DHC, the SHC will never be transmitted

(Ref: EIA 600.42, section 2.3.6)

Packet Data

Up to 32 bytes of packet data may be passed down to be transmitted.

Waiting for an MA_DATA.confirm [Packet_Confirm]

For every MA_DATA.request [Packet_Request] that is performed, an MA_DATA.confirm [Packet_Confirm] will be passed up to indicate whether the packet was transmitted successfully or not. Only two *Confirm_Results* exist to indicate a successful transmission (REMOTE_SUCCESS (0x0000) for Acknowledged packets, and LOCAL_SUCCESS (0xF000) for Unacknowledged packets), while many exist to indicate why the transmission was unsuccessful (see Appendix B for full explanations of these *Confirm_Results*.) Since only one transmit buffer exists within the PL-III, the host ***must*** wait for a confirm before transmitting another packet. The only exception to this is if the host wishes to abort the transmission of the current packet to send a higher priority request.

Note that the only association between an MA_DATA.request [Packet_Request] and its corresponding MA_DATA.confirm [Packet_Confirm] is a temporal one. MA_DATA.confirms [Packet_Confirm] will be passed up in the same order as the MA_DATA.requests [Packet_Request] were passed down.

Aborting a Packet / Transmitting a Higher Priority Packet

There may arise situations where it is necessary to abort the current transmission because another packet needs to be transmitted right away. In this case, an MA_DATA.request [Abort_Request] should be performed, which will abort the request that is currently in progress. Note that this does not abort a frame in the middle of a transmission, it only prevents any further retransmissions of the frame from occurring. The host must wait for the MA_DATA.confirm [Packet_Confirm] corresponding to this request before attempting to send the 2nd MA_DATA.request [Packet_Request] down. The *Confirm_Result* will probably be FAILURE_MACS_HIGHER_PRIORITY (0xF140), but could indicate that the previous packet was successfully transmitted, or was a failure for other reasons. If the packet was an ADR_UNACK_DATA packet, the *Confirm_Result* will be LOCAL_SUCCESS if the packet was transmitted at least once, even if it was not transmitted *Unack_Ch_Access_Num* number of times.

Parallel Interface

If the host is using the parallel interface, a 2nd method exists to abort the current packet.. If it is wished to send a higher priority request, a 2nd MA_DATA.request [Packet_Request] can be

performed before having received the confirm from the 1st MA_DATA.request [Packet_Request]. Once the 2nd request has been passed down, the MA_DATA.confirm [Packet_Confirm] for the first request will be passed up, and will have a value as described above.

When passing the 2nd MA_DATA.request [Packet_Request] down, there may be a long pause before the PL-III responds to the Primitive Type byte. This is because the PL-III may be in the middle of transmitting the packet already stored in its one transmit buffer. Allowing the 2nd MA_DATA.request [Packet_Request] to come down right away would overwrite the packet currently in the transmit buffer, and result in an invalid transmission on the medium. Therefore the PL-III waits until it is not transmitting on the channel before allowing the 2nd MA_DATA.request [Packet_Request] to continue coming down. The pause may be anywhere up to the time it takes to transmit one packet, which can be up to 70 ms for a full, worst-case packet. Typical packets will be transmitted in about 25 ms, with the typical pause being at most half of that, or 13 ms.

This procedure can be repeated *ad infinitum*, as soon as the MA_DATA.confirm [Packet_Confirm] for the packet that was aborted is received. In other words, a higher priority request can abort another higher priority request as soon as the MA_DATA.confirm [Packet_Confirm] for the original packet that was aborted is received.

Receiving a Packet

Once a packet has been received, the PL-III will pass it up to the host using the MA_DATA.indication primitive as soon as the interface is free. If the host only has one receive buffer, it is recommended that the *Automatic_Upper_Layer_Busy* bit in *CEWay_Configuration_Reg3* be set so that subsequent Acknowledged packets can be discarded and responded to with a REMOTE_BUSY confirm and so that subsequent Unacknowledged packets can be discarded without overwriting the host's single receive buffer. Once the host is capable of receiving another packet, it can indicate this to the PL-III via the LSM_EVENT.indication [UPPER_LAYER_NOT_BUSY]. If the host has more than one receive buffer, then the *Automatic_Upper_Layer_Busy* bit should probably not be set, and it is up to the host to send an LSM_EVENT.indication [UPPER_LAYER_BUSY] if it should run out of receive buffers.

Upper Layer Busy

The host may sometimes wish to indicate to the PL-III that it is too busy to accept any more packets. This can be accomplished via two methods:

- The host can set the *Automatic_Upper_Layer_Busy* bit in *CEWay_Configuration_Reg2* during its MA_SET_VALUE.request. This will result in the PL-III setting itself to UPPER_LAYER_BUSY whenever a packet addressed to it is successfully received. Once the packet has been passed up to the host, and the host is ready to receive new packets, it must indicate this to the PL-III by sending it the LSM_EVENT.indication [UPPER_LAYER_NOT_BUSY] primitive.

- The host can send an LSM_EVENT.indication [UPPER_LAYER_BUSY] to the PL-III. No packets will be passed up until the LSM_EVENT.indication [UPPER_LAYER_NOT_BUSY] primitive is sent down.

If the PL-III is UPPER_LAYER_BUSY, it will return FAILURE_REMOTE_BUSY Acknowledgements (0001 hex) to any Acknowledged packets it receives. Unacknowledged packets will simply be discarded.

Receive Duplicate List

Note that supporting the reception of an addressed service requires that an element be free in the Receive Duplicate List (Rx_Dup_List). This is a list maintained within the MACS used to distinguish new or duplicate packets received from other nodes. It is configurable via the lower nibble in *CEWay_Configuration_Reg3*.

Every time an addressed packet is received from an address not currently in the list, or which was sent using a different destination address, a new element in the Rx_Dup_List is used to maintain an association of a Sequence_Number bit with this new source address and its associated destination address (our local source address (House Code and Unit Address), one of our group addresses or one of the broadcast addresses). This association is maintained during a period of time equal to 1.25 * MAX_RETRANS_TIME (=0.9375 seconds), and then that element is freed up.

If an addressed packet is received from an address using an associated destination already in the list during that time, MACS compares the Sequence_Number bit of the newly received packet with the one from its Rx_Dup_List. If they match, it indicates that the newly received packet is a duplicate. This packet is rejected and the Rx_Dup_List element's timer is reset. If they do not match, it indicates that they are different and that the newly received packet is a new packet. Then the element in the list is re-used, the 0.9375 second timer is reset and the Sequence_Number bit is toggled.

If an addressed packet is received from a new address or using a different destination, but no elements are free within the Rx_Dup_List, then the packet is discarded and a FAILURE_REMOTE_BUSY_LIST (0x0010) Acknowledgement is sent only if the received packet used an Acknowledged service.

(Ref: EIA 600.41, section 3.2.5. EIA 600.42, sections 3.5.16 and 3.5.17.)

Other LSM Indications

Statistical_Counters

The PL-III can be configured (via *CEWay_Configuration_Reg3* during the MA_SET_VALUE.request) to save statistics on certain events. If these counters are enabled, then their values can be read by issuing an MA_READ_VALUE.request [STATISTICAL_COUNTERS]. The PL-III will respond with an MA_READ_VALUE.confirm [STATISTICAL_COUNTERS] containing the values of the 8 statistics that the PL-III logs.

Alternatively, if the *Support_Statistical_Counters_Overflow* bit is set in CEWay_Configuration_Reg2, each time a statistical counter overflows, an MA_EVENT.indication [STATISTICAL_COUNTER_OVERFLOW] will be passed up to the host. This indication contains one byte of data consisting of flags for each statistical counter. Those that just overflowed will have their flags set in this byte, and the counters will restart from zero.

Jabber Detection

The CEWay PL-III has built-in jabber detection abilities, as required by the CEBus standard. If the PL-III detects over 1000 consecutive Superior states in transmission, it resets its Physical Layer and indicates this condition to the host via the PH_EVENT.indication [PH_TX_JABBER_DETECT] LSM indication. Upon receiving this indication, the host should immediately reset the PL-III via its RESET* pin, and then wait for a minimum of 10,000 USTs (1 second) before continuing with the initialization process. This procedure is necessary to conform with the CEBus standard.

The PL-III can also detect jabber conditions in reception (>1000 consecutive Superior states on the channel), and will indicate this to the host via the PH_EVENT.indication [PH_RX_JABBER_DETECT] primitive, if the host has configured the MACS with the *Support_Diagnostic_Indication* bit set.

APPENDIX A

PROHIBIT ILLEGAL AND EXTENDED SERVICE BITS

Table 14 Prohibit Illegal and Extended Service Bits

Received Control Field			PROHIBIT ILLEGAL SERVICES	PROHIBIT EXTENDED SERVICES
Extended		*Basic*		
Single Byte	**Multiple Byte**	**Multiple Byte**		
OK	OK	OK	0	0
Remote Reject	Remote Reject	OK	0	1
OK	Bad Frame	Bad Frame	1	0
Bad Frame	Bad Frame	Bad Frame	1	1

Ok: Received frame passed up and Acknowledged as a success if ACK_DATA or ADR_ACK_DATA was detected in the 1st byte of control field.

Remote Reject: Received frame is not passed up, but a Failure_Remote_Reject_Extended (0x0090) is returned if ACK_DATA or ADR_ACK_DATA was detected in the 1st byte of control field. An MA_EVENT.indication [RX_OVERRUN] is passed up if this option is elected.

Bad Frame: Received frame is not passed up and is not Acknowledged. An MA_EVENT.indication [RX_OVERRUN] is passed up if this option is selected.

Additional Functions of the PROHIBIT ILLEGAL SERVICES Bit

Table 15 Additional Functions of the *Prohibit_Illegal_Services* Bit in Reception

Reception		
Case	**Action if bit = 1**	**Action if bit = 0**
Received a frame with the priority bits in the control field equal to Reserved (11).	Returns a Failure_Remote_Reject (if Acknowledged) and discards it.	The frame is successfully Acknowledged and passed up.
Received a frame with the Reserved bit in the control field set.	Returns a Failure_Remote_Reject (if Acknowledged) and discards it.	The frame is successfully Acknowledged and passed up.
An ADR_IACK is received and the sequence number is different than the one transmitted.	The ADR_IACK is ignored.	The ADR_IACK is accepted.

Reception		
Case	**Action if bit = 1**	**Action if bit = 0**
Received a multicast Acknowledged frame (ADR_ACK_DATA, ACK_DATA or ADR_IACK).	The frame is ignored.	The frame is passed up and a IACK or ADR_IACK is returned for Data frames.
Received a Broadcast House Code (0x0000) packet, with something else than the Broadcast Unit Address (0x0000).	The frame is ignored.	The frame is passed up and an Acknowledgement is returned for Acknowledged Data frames.

Table 16 Additional Functions of the *Prohibit_Illegal_Services* Bit in Transmission

Transmission		
Case	**Action if bit = 1**	**Action if bit = 0**
A frame is passed down to be transmitted with priority set to Reserved (11).	Returns a Failure_Macs_Illegal_Services _Priority confirm and the frame is not transmitted.	The Priority bits are set to “11” and the priority channel access delay is set to the same value as the DEFERRED priority.
A frame is passed down to be transmitted with the *DLL_Service_Level Reserved* bit set.	The MACS Control field Reserved bit is NOT set but the frame is transmitted.	The MACS Control field Reserved bit is set and the frame is transmitted.
A frame is transmitted with wrong broadcast address (DA != 0x0000 & DHC = 0x0000) or broadcast with Acknowledgement services (DA = 0x0000 or DHC = 0000 or GA range LOW <= DA <= GA range HIGH).	Returns a Failure_Macs_Illegal_Services confirm and the frame is not transmitted.	The frame is transmitted.

Appendix B

Explanation of Confirm Codes

Name: REMOTE_SUCCESS
Code: 00000H
Services: ACK_DATA, ADR_ACK_DATA
Explanation: An IACK, or an ADR_IACK with a null data field was received from the remote node.

Name: REMOTE_BUSY RANGE
Code: 00001H – 0007FH
Services: ACK_DATA, ADR_ACK_DATA
Explanation: The receiving node was unable to accept the message for some reason. A likely cause would be that all of its receive buffers were full.

Name: REMOTE_BUSY_RCV_LIST
Code: 00010H
Services: ADR_ACK_DATA
Explanation: The receiving node's Receive Duplicate List is full, and temporarily cannot receive the packet.

Name: REMOTE_BUSY_LIST_RESET
Code: 00020H
Services: ADR_ACK_DATA
Explanation: The receiving node's Receive Duplicate List is unable to be used, since the node has been recently (< 0.9375 s) reset (see *Support_Busy_After_Reset* in MA_SET_VALUE.request).

Name: REMOTE_REJECT RANGE
Code: 00080H – 000FFH
Services: ACK_DATA, ADR_ACK_DATA
Explanation: The receiving node cannot accept the received packet. A likely cause would be that the received frame had a Reserved bit set.

Name: REMOTE_REJECT_EXTENDED
Code: 00090H
Services: ACK_DATA, ADR_ACK_DATA
Explanation: The receiving node cannot accept packets using an EXTENDED service (ie. With the Service_Class bit in the control field set to 1).

Name: REMOTE_OTHER RANGE
Code: 00100H – 0DFFFH
Services: ACK_DATA, ADR_ACK_DATA
Explanation: These confirm codes are not defined yet.

Name: REMOTE_OTHER_OUT_OF_RANGE
Code: 0E000H
Services: ACK_DATA, ADR_ACK_DATA
Explanation: A remote confirm was returned between 0xE000 and 0xFFFF, which is out of the allowed range for remote confirms.

Name: LOCAL_SUCCESS
Code: 0F000H
Services: UNACK_DATA, ADR_UNACK_DATA
Explanation: The packet was successfully transmitted at least once. There is no way to know whether it was successfully received by the remote node(s).

Name: FAILURE_NO_ACKNOWLEDGE RANGE
Code: 0F110H – 0F11FH
Services: ACK_DATA, ADR_ACK_DATA
Explanation: The packet was transmitted at least once (plus the optional IRetry), but no IACK, FAILURE, or ADR_IACK was received.

Name: FAILURE_NO_ACKNOWLEDGE
Code: 0F110H
Services: ACK_DATA
Explanation: The packet was transmitted once (plus the optional IRetry), but no IACK or FAILURE was received within 6 USTs of transmission.

Name: FAILURE_NO_ACKNOWLEDGE_BAD_FRAME
Code: 0F111H
Services: ACK_DATA
Explanation: An improperly formatted frame was received within the IACK time slot.

Name: FAILURE_NO_ACKNOWLEDGE_RETRY_BACKOFF
Code: 0F112H
Services: ACK_DATA
Explanation: A collision occurred during the IRetry, however the packet received was not a properly formatted IACK

Name: FAILURE_NO_ACKNOWLEDGE_WRONG_TYPE
Code: 0F113H
Services: ACK_DATA
Explanation: A valid frame was received within the IACK time slot, but was not an IACK or FAILURE.

Name: FAILURE_NO_ACKNOWLEDGE_EXPIRED
Code: 0F114H
Services: ADR_ACK_DATA
Explanation: The Retrans_Timer expired without receiving an ADR_IACK.

Name: FAILURE_NO_ACKNOWLEDGE_ADR_IACK
Code: 0F115H
Services: ACK_DATA
Explanation: An ACK_DATA packet was sent, but an ADR_IACK was received (instead of the expected IACK).

Name: FAILURE_NO_ACKNOWLEDGE_HIGH_PRIORITY
Code: 0F116H
Services: ACK_DATA, ADR_ACK_DATA
Explanation: The packet was successfully transmitted but no Acknowledgement was received and has been replaced by a new packet from the host (Higher Priority request).

.

Name: FAILURE_EXCESSIVE_COLLISIONS RANGE
Code: 0F120H – 0F12FH
Services: ALL
Explanation: The packet was not transmitted a single time, probably due to excessive traffic on the channel.

Name: FAILURE_EXCESSIVE_COLLISIONS
Code: 0F120H
Services: UNACK_DATA, ACK_DATA
Explanation: The *Max_Restart* counter expired without the packet being transmitted a single time. RETRANS_TIME did not expire.

Name: FAILURE_EXCESSIVE_COLLISIONS_EXPIRED
Code: 0F123H
Services: ALL
Explanation: The RETRANS_TIME expired without the packet being transmitted a single time.

Name: FAILURE_EXCESSIVE_COLLISION_ACCESS_NUM
Code: 0F124H
Services: ADR_UNACK_DATA
Explanation: The *Unack_Ch_Access_Num* was set to 0 so no transmission was performed.

Name: FAILURE_MACS_TX_LOCAL_BUSY RANGE
Code: 0F130H – 0F13FH
Services: ADR_UNACK_DATA, ADR_ACK_DATA
Explanation: The packet could not be transmitted due to a problem with the Tx_Dup_List, most likely that it is full.

Name: FAILURE_MACS_TX_LOCAL_BUSY_LIST

Code: 0F131H
Services: ADR_UNACK_DATA, ADR_ACK_DATA
Explanation: The Tx_Dup_List is full.

Name: FAILURE_MACS_TX_LOCAL_BUSY_EXPIRED_LIST
Code: 0F132H
Services: ADR_UNACK_DATA, ADR_ACK_DATA
Explanation: The Tx_Dup_List is full and MACS has waited its full Retrans_Time (see *Wait_While_Tx_Busy_List* in MA_SET_VALUE.request).

Name: FAILURE_MACS_TX_LOCAL_BUSY_RESET
Code: 0F133H
Services: ADR_UNACK_DATA, ADR_ACK_DATA
Explanation: You cannot use Addressed services following any PL-III reset for 1.125 second if you are configured to *Support_Busy_After_Reset* (see *CEWay_Configuration_Reg2* in the MA_SET_VALUE.request [Configuration] primitive).

Name: FAILURE_MACS_HIGHER_PRIORITY RANGE
Code: 0F140H – 0F14FH
Services: ALL
Explanation: The packet was never successfully transmitted and has been aborted at the request of the host for a Higher Priority request.

Name: FAILURE_MACS_HIGHER_PRIORITY
Code: 0F140H
Services: ALL
Explanation: The packet was never successfully transmitted and aborted and may have been replaced by a new packet from the host (see Aborting a Packet / Transmitting a Higher Priority Packet).

Name: FAILURE_MACS_ILLEGAL_SERVICES RANGE
Code: 0F150H – 0F15FH
Services: ALL
Explanation: An illegal packet has been passed down to the MACS, and was not transmitted. The *Prohibit_Illegal_Services* bit was probably set.

Name: FAILURE_MACS_ILLEGAL_SERVICES
Code: 0F150H
Services: ALL
Other: *Prohibit_Illegal_Services* bit must be set.
Explanation: There are a couple of conditions where the MACS can return this code. They are as follows:

1) DHC = 0x0000 and DA != 0x0000, using any service. If sending a packet to the system broadcast address, the unit address must also be 0x0000.

2) ACK_DATA or ADR_ACK_DATA to broadcast or group address. You may not send Acknowledged packets to addresses where more than one node might receive them. This includes the broadcast address (DHC = 0x0000) or the group address range (0x0101 - 0x0FFF).

Name: FAILURE_MACS_ILLEGAL_SERVICES_PRIORITY
Code: 0F151H
Services: ALL
Other: *Prohibit_Illegal_Services* bit must be set.
Explanation: A packet with an illegal two-bit priority of "11" was passed to the MACS.

Name: FAILURE_MACS_ILLEGAL_SERVICES_XMIT_ADR
Code: 0F152H
Services: ADR_ACK_DATA, ADR_UNACK_DATA
Explanation: A configuration with no Tx_Dup_List elements was selected, and a packet using an Addressed service was passed to the MACS. Note that this confirm does not depend on the state of the *Prohibit_Illegal_Services* bit.

Appendix C

References

Many references are made throughout this data sheet to the EIA 600 series of documents, which specify the CEBus standard. These documents are available through Global Engineering, and can be obtained as follows:

On-line:

http://www.cebus.org/global.html
global@ihs.com

Off-line:

Global Engineering Documents
15 Inverness Way East
Englewood, COLORADO, 80112 ,USA
Phone: 1-800-854-7179
Fax: 1-303-397-2740

IMPORTANT NOTICE

DOMOSYS Corporation ("DOMOSYS") reserves the right to make changes to its products or services without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

DOMOSYS warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with DOMOSYS' standard warranty. Testing and other quality control techniques are utilized to the extent DOMOSYS deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

DOMOSYS PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of DOMOSYS products in such applications is understood to be fully at the risk of the customer. Questions concerning potential risk applications should be directed to DOMOSYS. In order to minimize risks associated with the customer applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

DOMOSYS assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does DOMOSYS warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of DOMOSYS covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Change Tracking

The following changes have taken place in this document

Date	Responsible	Description of change	Revised page(s)
Feb.19, 1998	H. Dufour	Pin out 9 was RSVD and is now NC.	1
Feb.19, 1998	H. Dufour	Pin out 25 was VDD and is now NC.	1
Feb.19, 1998	H. Dufour	Block Diagram: the pin CTRL1/RxD is now CTRL1/TxD	2
Feb.19, 1998	H. Dufour	The last sentence was added to the opening paragraph.	14
Feb. 24, 1998	A. Dunn	Added Documentation: paragraph on Test Mode.	8
Feb. 24, 1998	A. Dunn	Rx_Error_Frame_Count was modified to include CRC errors.	33
Mar. 5, 1998	A. Dunn	Description of Support_Busy_After_Reset was modified, and now describes how Unack_Data packets are not passed up.	33
Mar. 13, 1998	A. Dunn	Description of Support_Busy_After_Reset was modified again, and notes that no packets are passed up for the full duration of the Rcv_Dup_Timer. The suggested value for this feature was also changed from 1 to 0.	33
May 13, 1998	H. Dufour	In Timing Diagrams: CEWaystr* changed to CEWstr*	13
Oct 6, 1998	A. Dunn	Got rid of 38,400 baud as possible host communication baud rate.	16
Dec 22, 1998	A. Dunn	Added notes about need for pull-up resistors on CEWstr, CTRL0, CTRL1, and I/O 0-7 pins for some designs.	3, 10
March 200	A.G.	Correction to output voltage	4

Domosys Corporation
1995, rue Jean-Talon Sud
Suite 202
Sainte-Foy, Quebec
Canada G1N 4H9
Tel: (418) 681-8022
Fax: (418) 681-8015
e-mail: support@domosys.com
www.domosys.com