

LynX-10TM COPROCESSOR
Bi-directional RS-232 to X-10 Controller
Model: 103 Enclosed Unit

Marrick Limited LynX-10TM Coprocessor Manual revision 2.00
Marrick Limited, Incorporated
P.O. Box 950940
Lake Mary, FL 32795
(800) 928-5299
(407) 324-1291 FAX
EMAIL: sales@marrickltd.com
Website: www.marrickltd.com

Copyright © 1995-1998 Marrick Limited

Table of Contents

1.0 Overview of X-10 Technology.....	5
2.0 Technical Overview	
2.1 Introduction to a LynX-10™ Session.....	6
2.2 LynX-10™ Protocol.....	7
2.3 Mapping X-10 units to LynX-10™ Codes.....	7
2.4 LynX-10™ Protocol lookup Tables.....	8
2.4.1 LynX-10™ Protocol Lookup Table Notes.....	8
3.0 Command Summaries	
3.1 ON Command Summary.....	9
3.1.1 ON Command Notes.....	9
3.2 OFF Command Summary.....	9
3.2.1 OFF Command Notes	9
3.3 DIM Command Summary.....	10
3.3.1 DIM Command Summary (Mode 0).....	10
3.3.2 DIM Command Summary (Mode 1).....	10
3.3.3 DIM Command Notes.....	10
3.4 X-10 STATUS Command Summary.....	11
3.4.1 X-10 STATUS Command Notes.....	11
3.5 RESET Command Summary.....	11
3.6 TIMER REQUEST Command Summary.....	11
3.6.1 TIMER REQUEST Command Notes.....	11
3.7 DIRECT X-10 ACCESS Command Summary.....	12
3.7.1 DIRECT X-10 ACCESS Command Notes.....	13
3.8 X-10 HAIL Command Summary.....	14
3.8.1 X-10 HAIL Command Notes.....	14
3.9 LynX-10™ MICROCODE VERSION REQUEST Command Summary.....	15
3.10 WRITE TO EEPROM Command Summary.....	15
3.11 LynX-10™ STATISTICS COUNTER REQUEST Command Summary.....	16
3.11.1 STATISTICS COUNTER REQUEST Command Notes.....	16
3.12 LynX-10™ REGISTER ACCESS Command Summary.....	17
3.12.1 REGISTER ACCESS Command Notes.....	17

4.0	Error Codes	
4.1	LynX-10™ Coprocessor Error Codes.....	18
4.1	LynX-10™ Coprocessor Host Error Codes.....	18
5.0	Hardware Considerations	
5.1	Communications.....	19
5.2	Hand Shaking.....	19
5.3	Power Supply.....	19
6.0	Software Considerations	
6.1	MODE Register.....	20
6.1.1	DEBUG Mode.....	21
6.1.2	Return String Terminators.....	21
6.1.3	50/60 Hz Automatic Detection and Switching.....	21
	Appendix A - Indicators and Connectors	22
	Appendix B - Hexadecimal Numbering Overview.....	23
	ASCII Table.....	24

1.0 OVERVIEW OF X-10 TECHNOLOGY

X-10 control is one of the most prolific and low cost means to automate a home or business, especially if running new wiring is prohibitive. The only major drawback to the system is the computer interface. There have been many attempts to provide a reliable two-way connection from the computer or computer like equipment to the X-10 world. The first effort was by the Power House (X-10) people themselves. This was the TW523 module which provided the optically isolated connections to the power line. It did not provide the timing required to generate the correct control commands. Next came several simple interfaces that were merely level translators to the TW523 to hook into the computer's serial or parallel ports. These interfaces required the computer to perform all the monitoring and timing requirements greatly loading the computer system. Today with the advent of multitasking systems such as Microsoft Windows™ and UNIX™, the computer cannot service a single task for 1 or 2 seconds. This would cause disastrous results in the software. Next came the chips! This was the first attempt to free the computer from the burden of the X-10 timing. These chips have parallel interfaces on one side and the TW523 interface on the other. They also suffer from several problems in that the device still needs to be polled very rapidly so as not to miss any incoming X-10 signals. X10 signals can appear at any zero crossing of the 60 Hz power or every 16.67 mS.

The Marrick Limited solution was to go back to ground zero. We designed a complete TW523 (X-10) compatible controller and interface into a single chip. The LynX-10™ Coprocessor off loads the computer of the complex timing and constant polling to ensure data integrity. This board provides an asynchronous RS-232 level serial 1200 bps interface on one side, and the TW523 compliant interface on the other. *ALL the timing and monitoring is performed by this board.* It also has many built in commands to automatically handle Dimming, Brightening and Broadcast messaging. This version also provides a 32 BYTE receive FIFO on the serial port, X10 collision detection and auto retransmission, a debug mode, and many, many new and exciting features.

2.0 TECHNICAL OVERVIEW

2.1. INTRODUCTION TO A LynX-10™ SESSION

The computer interface uses ASCII commands to carry out its functions. These commands are outlined below. The basic structure is a letter command such as 'N' for ON, followed by data to determine the house code, address, and type of action (all on, one unit, etc.). For example, if you wanted to turn on unit A4, you would send the ASCII string 'N003'. The controller would respond with an asterisk (*) when completed, or an 'E' if there was an error followed by a number to further explain the error. All return strings are terminated with an optional "Return String Terminator" (RST). The default RST is the carriage return character (ASCII 0x0D). Here is a typical X-10 session...

```
COMPUTER:      N000      ; Turn on unit A1
CONTROLLER:    *          ; Done... (asterisk is followed by the programmed return string)
COMPUTER:      D705      ; Dim unit A6 to level 7 (mode 0)
CONTROLLER:    *          ; Done...
CONTROLLER:    X004      ; Controller reports X-10 Received (HC=A, Unit 5)
CONTROLLER:    X102      ; Controller reports X-10 Received (HC=A, On)
COMPUTER:      F21       ; Turn off all units, house code B.
CONTROLLER:    *          ; Done...
```

Look at the last command from the computer. It instructed the controller to send an X-10 ALL UNITS OFF command to house code B. There was no need to send the address of the unit since the command applies to all addresses on that house code. This is true of all commands that address more than one unit. RESET ('R') for instance does not require any other byte to reset the controller. If the controller needs more data, the green 'BUSY' LED will stay lit until enough bytes have been received to satisfy the command requested. The first byte is the COMMAND byte, the next byte (if required) is the TYPE byte and is used to further define the action. The next byte is the HOUSE CODE byte (ASCII letters for 0-F HEX) and describes house codes A-P. The last byte (if required) is the ADDRESS byte (0-F HEX) that describes the unit address. The exception for the address byte is when sending raw X-10 commands with the 'X' command (also receiving). If byte following the 'X' is '0', then the last byte is the address. If the byte following the 'X' is a '1', then the last byte is an X-10 command code such as ALL UNITS ON. See the above example. Following is an overview of all the commands and their extensions.

2.2. LynX-10™ PROTOCOL

The LynX-10™ Coprocessor uses a condensed method for addressing and reporting unit codes. Since there are exactly 16 house codes (A-P) and 16 unit codes (1-16), it is convenient to use a hexadecimal equivalent mapping. For those not completely familiar with the hexadecimal number system, refer to Appendix C. Each house code is mapped to a hex number to allow only 1 digit per house code and 1 digit for unit code address.

Commands can take several forms, but most follow these simple rules.

1. All characters sent to the LynX-10 Coprocessor are printable ASCII codes. That is, you can send them to a terminal or printer and see the letter or number. The actual value of the code is quite different. For example, the letter “A” is ASCII code 0x41 in hexadecimal or 65 decimal. The letter “A” in our context stands for 10 decimal. One exception is the carriage return character, used in some commands, and the return string characters returned after an X-10 reception. See MODE register for details on return string terminators (RST).
2. The code structure always begins with a command code. These codes are listed in table 3 along with examples of their structure. For example, to turn on a single unit the command is “N0xy” where “xy” are found in table 1 from the house and unit code of the module you desire to turn on. If you wanted to turn on unit C3, the string of characters would be “N022”
3. Some commands use both the house code and unit code (or key code), and some only use the house code. For commands that activate every unit on a particular house code, such as ALL UNITS ON, only the house code is needed. These commands take a form similar to above, but with one less digit. For example, to turn on all units this house code for house code A, the string would be “N10” where the “1” is ALL UNITS this house code, and the “0” is for house code A from table 2.

2.3. MAPPING X-10 UNITS TO LynX-10™ CODES

To make interfacing to computers easier, a special code system is used for addressing specific units and house codes. In table 1, every possible code that can be set on an X-10 module is mapped to its hexadecimal value used with the LynX-10™ protocol. For example, if you want to address unit E10, you would look up the letter “E” along the left axis, and the number “10” on the top axis of table 1. Where the two columns intersect are the two characters used with the LynX-10™ Coprocessor to address that unit. If a command addresses an entire house code, then table 2 is used. This maps the house code to a single character for the LynX-10 protocol.

2.4. LynX-10™ PROTOCOL LOOKUP TABLES

Table 1 - Unit code to LynX-10™ protocol cross reference

UC HC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
B	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
C	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
D	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
E	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
G	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
H	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
I	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
J	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
K	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
L	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
M	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
N	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
O	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
P	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

(HC = House code, UC = Unit code -> mapping to 2 ASCII printable characters)

Table 2 - X-10 house codes mapped to LynX-10™ protocol

X-10	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
LynX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

2.4.1 LynX-10™ Protocol Lookup Table Notes

The programmer should use the tables above to determine the actual characters sent to address a unit within a command. For instance, if the command requires a house code and a unit code such as the ON command “N0”, use table 1 to look up the two characters that follow the “N0” to address the unit. Let’s say we want to address unit M15 for the ON command. We would go to table 1 and find the house code M on the left and the unit number or unit code 15 on the top. Tracing the intersection of these two we would find the two characters “CE” which is a character string representing the hexadecimal value of 0xCE. Therefore the entire ON command string sent to the coprocessor would be represented by “NOCE”. That’s the letter “N” followed by the number “0” (zero), followed by the letter “C” followed by the letter “E”. If only the house code is required as with the “N1” command, table 2 is used. The house code is on the top, and the corresponding code is on the bottom. For house code H, the character would be the number “7”, so the whole command string would be “N17”.

3.0 COMMAND SUMMARIES

3.1 ON Command Summary

Command	Description	Example / Comments
N0xy	Turn on single unit xy (xy from table 1)	“N0F0” turns on unit P1
N1x	Turn on all units this house code (x from table 2)	“N15” turns on all units on house code F
N2	Turn on all units, all house codes	“N2” turns on all units on every house code
N3-NF	(Reserved)	Returns error E1

3.1.1 ON Command Notes

The “N2” command is very useful for PANIC button implementations. With one command to the LynX-10™ Coprocessor, every unit on every house code in the house (or your neighbor’s house!) will turn on. This is a great way to scare an intruder, since it carries this out in about 6 seconds. Light is a great deterrent to intruders, since they usually try hard to avoid identification.

3.2 OFF Command Summary

Command	Description	Example / Comments
F0xy	Turn off single unit xy (xy from table 1)	“F03F” turns off unit D16
F1x	Turn off all lights house code x (x from table 2)	“F12” turns off all lights on house code C
F2x	Turn off all units house code x (x from table 2)	“F29” turns off all units on house code J
F3	Turn off all lights, all house codes	“F3” turns off all lights on every house code
F4	Turn off all units, all house codes	“F4” turns off all units on every house code
F5-FF	(Reserved)	Returns error E1

3.2.1 OFF Command Notes

The OFF commands have several flavors. Since there is a distinction between lights and units, there are OFF commands for each. When addressing a single unit, it is immaterial which it is. But when addressing groups of units, there are commands that affect only the light modules, and some that affect all modules. Marrick Limited has done some experimentation and found that some units will only respond to the ALL UNITS OFF code such as appliance modules and some light modules. These can be addressed individually, but when an ALL LIGHTS OFF command is sent, they do not turn off. Only the ALL UNITS OFF command will turn them off. You may want to experiment with your modules to see how they respond.

3.3. DIM Commands

3.3.1 DIM Command Summary (MODE 0)

Command	Description	Example / Comments
Dnxy	Dims unit <i>xy</i> to level <i>n</i> (<i>xy</i> from table 1, <i>n</i> from table 3)	“DB00” Dims unit A1 to level 11

3.3.2 DIM Command Summary (MODE 1)

Command	Description	Example / Comments
D0xn	Sends <i>n</i> number of DIMs to house code <i>x</i> (<i>x</i> from table 2, <i>n</i> from table 4)	“D035” sends 6 DIMs to house code D
D1xn	Sends <i>n</i> number of BRIGHTs to house code <i>x</i> (<i>x</i> from table 2, <i>n</i> from table 4)	“D124” sends 5 BRIGHTs to house code C
D2-DF	(Reserved)	Returns error E1

	Low				Med								Hi			
Level	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Table 3 - Dim levels for DIM command (n for MODE 0 DIM command)

Count	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Table 4 - Number of DIMs or BRIGHTs (n for MODE 1 DIM command)

(Note: In DIM MODE 1, sending a count of 0 sends 1 DIM or BRIGHT command)

3.3.3 DIM Command Notes

The DIM command can run in either of two modes defined by BIT 1 of the MODE register. MODE 0 is the legacy or original mode where an absolute level is specified. MODE 1 sends up to 16 DIMs or BRIGHTs in a row. This is useful for BRIGHT up from DIM.

3.4. X-10 STATUS Command Summary

Command	Description	Example / Comments
S0x	Sends a status OFF command for the unit requested (x from table 2) note: see text	“S0F” sends OFF status for house code P unit
S1x	Sends a status ON command for the unit requested (x from table 2) note: see text	“S11” sends ON status for house code B unit
S2xy	Requests status of unit xy (xy from table 1) note: see text	“S2CA” requests current status for unit M11
S3-SF	(Reserved)	Returns error E1

3.4.1 X-10 STATUS Command Notes

These codes were defined by Power House to check the status of individual units. There is only one known unit from them that uses it. This is the model number PR511 outdoor floodlight. This unit will respond with a STATUS OFF or STATUS ON code when addressed and its status requested. Other Marrick Limited products such as our LynX-PORT™ X-10 compatible input/output board uses these status commands.

3.5 RESET Command Summary

Command	Description	Example / Comments
R	Resets LynX-10™ Coprocessor	“R” Clears “SINCE RESET” timer (see below). Reloads settings from EEPROM

3.6 TIMER REQUEST Command Summary

Command	Description	Example / Comments
T	Request time since LynX-10™ Coprocessor reset. Returned as DD:HH:MM:SS where DD is number of days 00-99, HH is hours 00-23, MM is minutes 00-59, and SS is seconds 00-59.	“T” reports time since power line carrier present or reset.

3.6.1 TIMER REQUEST Command Notes

The timer can be used to determine if the carrier has returned after an E7 (carrier lost) error is received at the host. The timer will always remain at 00:00:00:00 until the power at the TW-523 returns. The programmer can check every several seconds or so to see if this timer has started incrementing again. This would indicate the return of the power at the TW-523 module. Since this is the only command that returns a number first, it can be easily parsed by checking for either a colon (“.”) or a number without a command identifier such as “X” or “E”.

3.7 DIRECT X-10 ACCESS Command Summary

Command	Description	Example / Comments
X0xy	Addresses unit <i>xy</i> for next command (<i>xy</i> from table 1)	“X067” address X-10 unit G8
X1x0	Sends an ALL UNITS OFF command to house code <i>x</i> (<i>x</i> from table 2)	“X1F0” sends ALL UNITS OFF to house code P
X1x1	Sends an ALL LIGHTS ON command to house code <i>x</i> (<i>x</i> from table 2)	“X141” sends ALL LIGHTS ON to house code E
X1x2	Sends an ON command to house code <i>x</i> (<i>x</i> from table 2)	“X102” sends ON to house code A
X1x3	Sends an OFF command to house code <i>x</i> (<i>x</i> from table 2)	“X113” sends OFF to house code B
X1x4	Sends a single DIM to house code <i>x</i> (<i>x</i> from table 2)	“X154” sends DIM to house code F
X1x5	Sends a single BRIGHT to house code <i>x</i> (<i>x</i> from table 2)	“X155” sends BRIGHT to house code F
X1x6	Sends ALL LIGHTS OFF command to house code <i>x</i> (<i>x</i> from table 2)	“X106” sends ALL LIGHTS OFF to house code A
X1x7	Sends EXTENDED CODE to house code <i>x</i> (<i>x</i> from table 2) note: see text	“X127” sends EXTENDED CODE to house code C
X1x8	Sends HAIL REQUEST to house code <i>x</i> (<i>x</i> from table 2) note: see text	“X1B8” sends HAIL REQ to house code L
X1x9	Sends HAIL ACKNOWLEDGE to house code <i>x</i> (<i>x</i> from table 2) note: see text	“X1B9” sends HAIL ACK from house code L
X1nA	Sends PRESET DIM with MSB of 0 and <i>n</i> level (<i>n</i> from table 3) note: see text	“X15A” commands last addressed light to level 5
X1nB	Sends PRESET DIM with MSB of 1 and <i>n</i> level (<i>n</i> from table 3) note: see text	“X12B” commands last addressed light to level 18
X1xC	Sends EXTENDED DATA to house code <i>x</i> (<i>x</i> from table 2) note: see text	“X10C” sends EXTENDED DATA to house code A
X1xD	Sends STATUS ON for house code <i>x</i> (<i>x</i> from table 2) note: see text	“X15D” sends STATUS ON to house code F
X1xE	Sends STATUS OFF for house code <i>x</i> (<i>x</i> from table 2) note: see text	“X15E” sends STATUS OFF to house code F
X1xF	Sends STATUS REQUEST to house code <i>x</i> (<i>x</i> from table 2) note: see text	“X12F” requests status from last addressed unit on house code C
X2-XF	(Reserved)	Returns error E1

3.7.1 Direct X-10 Access Command Notes

The direct access commands allow the programmer to have direct control over X-10 modules and subsystems. This provides the most flexibility, but also requires more attention to detail. Code X0 is used to address units, and can be used several times in a row on the same house code to address a group of units. The X1 command along with its subcommand is used to tell the unit or units what to do, or to request information from advanced X-10 units. If a group is addressed, only a single command is needed to cause all the units to respond. This is useful when turning on or off several units together. It is also quicker to do so in that the individual commands are not sent. So if seven units are addressed and then a single on command is sent, there will be a net savings of 6 commands or over 2 seconds.

Certain X-10 codes such as HAIL REQUEST or STATUS REQUEST require advanced X-10 modules to operate. Not all modules will respond to these commands. Ask the manufacturer for details on what commands are supported and how they are implemented. There are two commands called EXTENDED CODE and EXTENDED DATA. These commands can be transmitted with the LynX-10 Coprocessor board, but their documented implementation cannot do to the limits of the TW-523 interface module. These codes require a continuous data stream without the normal 3 power line cycle idle periods between them. The TW-523 will not receive these codes. Therefore, only the command itself without the data can be sent. This still can be utilized to activate remote equipment, or to tell a unit the next address it receives is data, not address. This would allow a “nibble” mode of data transfer between LynX-10 subsystems.

3.8 X-10 HAIL Command Summary

Command	Description	Example / Comments
H0	Requests if any other controller is using house code A. note: see text	“H0” requests control of house code A
H1	Requests if any other controller is using house code B. note: see text	“H0” requests control of house code B
H2	Requests if any other controller is using house code C. note: see text	“H0” requests control of house code C
H3	Requests if any other controller is using house code D. note: see text	“H0” requests control of house code D
H4	Requests if any other controller is using house code E. note: see text	“H0” requests control of house code E
H5	Requests if any other controller is using house code F. note: see text	“H0” requests control of house code F
H6	Requests if any other controller is using house code G. note: see text	“H0” requests control of house code G
H7	Requests if any other controller is using house code H. note: see text	“H0” requests control of house code H
H8	Requests if any other controller is using house code I. note: see text	“H0” requests control of house code I
H9	Requests if any other controller is using house code J. note: see text	“H0” requests control of house code J
HA	Requests if any other controller is using house code K. note: see text	“H0” requests control of house code K
HB	Requests if any other controller is using house code L. note: see text	“H0” requests control of house code L
HC	Requests if any other controller is using house code M. note: see text	“H0” requests control of house code M
HD	Requests if any other controller is using house code N. note: see text	“H0” requests control of house code N
HE	Requests if any other controller is using house code O. note: see text	“H0” requests control of house code O
HF	Requests if any other controller is using house code P. note: see text	“H0” requests control of house code P

3.8.1 X-10 HAIL Command Notes

The HAIL command was devised by Power House to enable automatic switching of house codes on simple, low module count systems. It was only implemented on a few systems and in general can be used for anything the programmer decides. The intent of the command was to send out a HAIL REQUEST on a house code to see if anyone else (like a neighbor) was using it. If another intelligent controller received the HAIL REQUEST on its house code, it would respond with a HAIL ACKNOWLEDGE. The initiating controller would then try another house code until an unused one could be isolated.

3.9 LynX-10™ MICROCODE VERSION REQUEST Command Summary

Command	Description	Example / Comments
V0	Returns current version number of LynX-10™ Coprocessor's microcode to host in the form Vxxx-xxx where xxx-xxx is the version.	"V0" returns current version to host.
V1	Returns copyright of LynX-10™ microcode to host. This is always "Copyright (c) xxxx Marrick Limited, Inc." where xxxx is the year of release.	"V1" returns current date of copyright to host.
V2-VF	(Reserved)	Returns error E1

Note: Sending carriage return after the "V" returns both the copyright and the version number.

3.10 LynX-10™ WRITE TO EEPROM Command Summary

Command	Description	Example / Comments
W0	Saves current settings into the EEPROM. This includes the MODE register, the SERIAL FIFO THRESHOLD register, and the X-10 RETRANSMISSION ATTEMPTS register.	"W0" will save all setting for next reset or power-on.
WF	Restores all factory settings to the LynX-10™ Coprocessor board. This will set the MODE register to 00, and set both the FIFO threshold and the X-10 retransmission attempts to 16 (10 hexadecimal). <i>It also will clear ALL statistics counters to 0 and reset the run timer to 0.</i>	"WF" restores all factory settings and resets the board.

NOTE: During power-up or reset, all settings are read from the EEPROM (if present) and restored into memory. If a temporary change is made and it is not desired to remain after reset, do not save the settings to the EEPROM.

3.11 LynX-10™ STATISTICS COUNTER REQUEST Command Summary

Command	Description	Example / Comments
C00 ↵	Returns value of statistics counter 00 in form C00=xxxx where xxxx is character string representing the HEX value of the count. This counter is for error E0 - X-10 RECEPTION ERRORS. NOTE: This counter can be cleared by sending the string "C00=0" to the LynX-10 board.	"C00" followed by a carriage returns to the host "C00=0045" which would indicate that 0x0045 X-10 reception errors have occurred since the counter was cleared.
C01 ↵	Same as above for errors E1 - BAD COMMAND FROM HOST.	(see above example)
C02 ↵	Same as above for errors E2 - BAD DATA FROM HOST.	(see above example)
C03 ↵	Same as above for errors E3 - X10 COLLISIONS DETECTED.	(see above example)
C04 ↵	Same as above for errors E4 - X10 TRANSMISSION FAILURE.	(see above example)
C05 ↵	Same as above for errors E5 - X10 LOST RECEPTION.	(see above example)
C06 ↵	Same as above for errors E6 - SERIAL RECEIVE BUFFER OVER-RUN.	(see above example)
C07 ↵	Same as above for errors E7 - LOST CARRIER (no power at TW-523)	(see above example)
C08 through CFF	(Reserved)	Returns error E2

3.11.1 STATISTICS COUNTER REQUEST Command Notes

The symbol ↵ represents a carriage return or ASCII 0x0D. Also, all counters can be cleared in the same fashion as C00 shown above. For example, to clear C07, send "C07=0" to the LynX-10™ Coprocessor. Counters C08 through CFF are reserved for future revisions of the microcode and will return an error E2.

3.12 LynX-10™ REGISTER ACCESS Command Summary

Command	Description	Example / Comments
M00 ↵ or M00=xx	<u>MODE REGISTER</u> M00 followed by a carriage return will return to the host the current value of the MODE register. M00=xx will set the MODE register to the hexadecimal value represented by the character string xx. NOTE: See MODE register for details	The MODE register is used to set the operation of the LynX-10™ Coprocessor. Example: “M00=01” will enable DEBUG mode by setting bit 0 of the MODE register.
M01 ↵ or M01=xx	<u>SERIAL FIFO THRESHOLD</u> M01 followed by a carriage return will return to the host the current value of the SERIAL FIFO THRESHOLD register. M01=xx will set the register to the hexadecimal value represented by the character string xx. This value can be anything between 01 and 1C.	The threshold is set from the factory at hexadecimal 10 or decimal 16. This puts it at the half way point of the FIFO for best results with 16550 UARTS. “M01=1A” would set it to decimal 26.
M02 ↵ or M02=xx	<u>X-10 RETRANSMISSION ATTEMPTS</u> M02 followed by a carriage return will return to the host the current value of the X-10 RETRANSMISSION ATTEMPT register. M02=xx will set the register to the hexadecimal value represented by the character string xx. This value can be anything between 01 and 1C. If an attempt is made to load any other value, the register is automatically adjusted up or down to keep it within these boundaries.	The X-10 retransmission attempt register is used to adjust the number of tries for the X-10 transmission routine after collisions occur. This is set to hexadecimal 10 or decimal 16 from the factory. It can be set anywhere from 00 (no attempts after collision) to FE (254 attempts). NOTE: Setting this register to FF will cause continuous attempts forever until a successful transmission occurs without a collision.
M03 through MFF	(Reserved)	Returns error E2

3.12.1 REGISTER ACCESS Command Notes

The symbol ↵ represents a carriage return or ASCII 0x0D. See section 6.1 regarding the functions of each bit in the MODE register. Also, be careful when setting the retransmission counter to “FF” or decimal 255 which will continue to transmit *FOREVER* or until the transmission is successful. This will prevent any other command from getting in front of it including the RESET command.

4.0 ERROR CODES

4.1 LynX-10™ COPROCESSOR ERROR CODES : (LEDS)

NOTE: ALL LEDS TURN ON FOR TEST, ALL GO OFF IF HEALTHY, ELSE...

1. ERROR LED ONLY : BAD ACCUMULATOR (FATAL)
2. ERROR & RX LED : BAD REGISTER (FATAL)
3. ERROR & TX LED : BAD MEMORY LOCATION (FATAL)

4.2 LynX-10™ COPROCESSOR HOST ERROR CODES:

NOTE: SENT TO HOST FROM COPROCESSOR

- E0 : RECEPTION ERROR (X10)
- E1 : BAD COMMAND RECEIVED FROM HOST
- E1 : BAD DATA RECEIVED FROM HOST
- E3 : X10 COLLISION DETECTED DURING TRANSMISSION
- E4 : X10 TRANSMISSION FAILURE (TIME OUT)
- E5 : X10 LOST RECEPTION (LOST TX ECHOES FROM TW-523)
- E6 : SERIAL COMMUNICATION RX FIFO OVER-RUN
- E7 : CARRIER LOST (50/60Hz POWER FAILURE AT TW-523)

NOTE: All error codes map to statistics counters one to one. Example: if an E4 error is received by the host, the statistics counter 04 (C04) will be incremented by 1.

5.0 HARDWARE CONSIDERATIONS

5.1 COMMUNICATIONS

The LynX-10™ Coprocessor communicates with the host at 1200 bits per second (bps). This may seem slow, but the actual symbol rate on the power line is only 60 symbols per second. It takes 11 symbols to complete 1 address or command and each group of 11 symbols are transmitted twice to assure reception. Most commands such as “ON” and “OFF” require an address (House code and unit ID) and a command (House code and command). Each are 11 symbols long including the start field. These are sent twice which means the actual rate at which things turn on or off is about 44 cycles of the 60 cycle power or over 2/3 of a second per 4 bytes of data from the host or about 22 bps. The full setup for the host’s serial port is:

1200 BPS, NO PARITY, 8 DATA BITS, 1 STOP BIT

5.2 HAND SHAKING

The MT200 LynX-10™ Coprocessor family has a 32 byte queue that holds transmitted data from the host. Once the command begins sending X-10 data on the power line, the coprocessor can continue to receive data until the FIFO threshold is reached. At this point, the coprocessor de-asserts Clear To Send (CTS) and will optionally send an XOFF (0x13) code to the host (see MODE register settings). When the level of the internal FIFO falls below the threshold level, the CTS signal is re-asserted and optionally the XON (0x11) code is sent to the host. From the factory, the threshold is set at 16 (half the size of the FIFO). This will allow a 16550 UART to flush it’s FIFO if the host serial driver software is emulating hardware handshaking without losing data. Optionally this threshold can be set anywhere between 1 and 31 bytes. We recommend leaving it set for 16 bytes for optimal performance and best communications integrity. **NOTE: If the internal FIFO is over-run, data will be lost and an E6 error will be sent to the host for each dropped character.**

Return data from the power line is decoded into raw X10 commands that all begin with an ‘X’ as described in the command section above and sent directly to the host with no handshaking. A receive queue must be large enough or serviced frequently enough to prevent overrun of receive data. Most communications drivers have sufficient queues including those for Windows, Windows NT, Windows 95, UNIX, OS/2, and DOS.

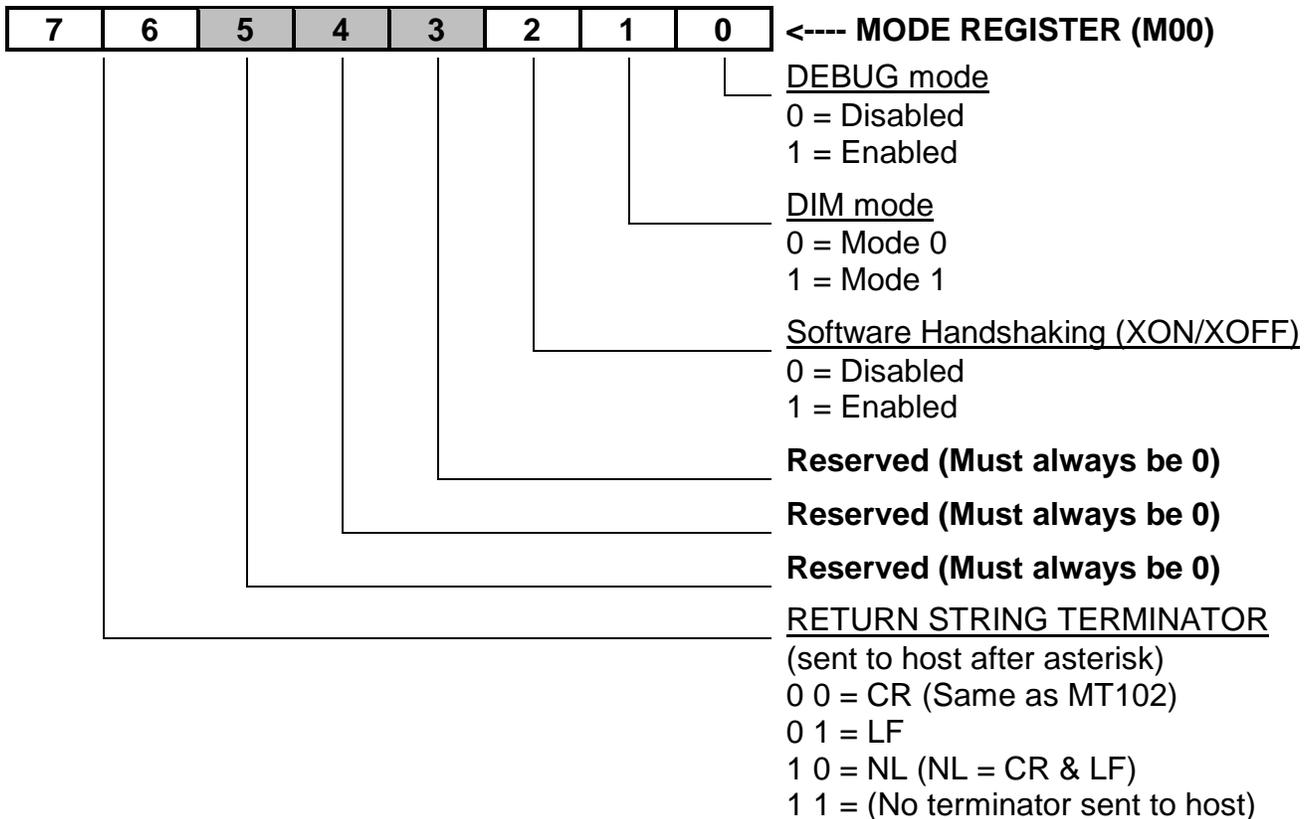
5.3 POWER SUPPLY

The Model 103 LynX-10 Coprocessor can accept input power at in the range of 10-25 volts AC or DC and consumes about 50mA of current. A power supply is included with the model 103 LynX-10 Coprocessor.

6.0 SOFTWARE CONSIDERATIONS

6.1 MODE REGISTER

The mode control register sets the programmable features of the MT200 series X10 controllers. This register can be copied to the EEPROM external to the controller. When the controller is powered up or reset, the settings saved in the EEPROM will be automatically reloaded. The usage of each bit is described below. To set these bits, the programmer must write an entire hexadecimal value to the mode register using the “M00=xx” command where xx is the hexadecimal byte. The mode register can be read back out by issuing a “M00” string followed by a carriage return (0x0D). For example if the programmer wishes to set bits 1 and 6, that would be a hexadecimal 42 (0x42). The string would be “M00=42”. This would enable DIM MODE 1, and set the return string terminator to a LINE FEED.



6.1.1 DEBUG Mode

The LynX-10™ Coprocessor can be put into a mode that will send both the X command and the raw bits received off the power line to the host. When DEBUG mode is enabled in the MODE register, any data received from the TW-523 will be sent to the host as raw bits followed by the X interpreted commands. The raw bits are prefaced by the letter “Y” to indicate the data type to allow software parsing. Example: Received data might look like this in DEBUG mode.

```
Y1110010101010101100101
X0C2
Y1110010101010101100110
X1C2
```

The characters after the “Y” represent the undecoded raw X-10 bits from the TW-523. A “1” character represents the presence of the 120khz signal on a zero crossing. A “0” character represents the absence of the 120khz signal. See the Power House TW-523 specification for details on how these bits are encoded. This mode is enabled by setting bit 0 of the MODE register (see MODE register above)

6.1.2 Return String Terminators

When a command is successfully completed, the coprocessor sends an asterisk (“*” - 0x2A) to the host. The MT200 LynX-10™ Coprocessor family has the ability to return several string termination characters following the asterisk. These are CR (carriage return - 0x0D), LF (line feed - 0x0A), NL (new line - CR & LF), or no string terminator. This allows the programmer to use the return string character that the operating system can handle. See the MODE register for details on how to set this up.

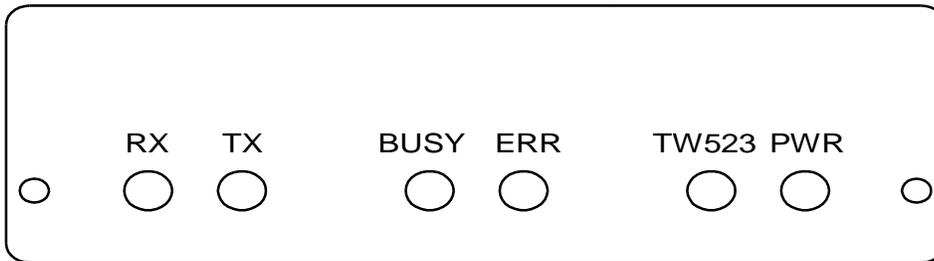
6.1.3 50/60Hz Automatic Detection and Switching

The LynX-10™ Coprocessor automatically detects the frequency of the line current. This is monitored every cycle of the power line and the internal timing is automatically adjusted to either 50 Hz (Europe) or 60 Hz (USA and Canada). No user intervention is required.

NOTE: The Power House TW-523 must be the correct version for the region of the world in which it will be used.

Appendix A – Indicators and Connectors

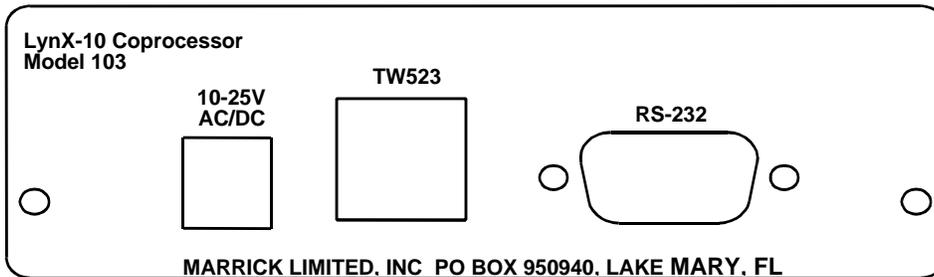
Front view of LynX-10 Coprocessor showing LED indicators



Indicator Description Table

INDICATOR	DESCRIPTION
RX	X-10 Receive activity
TX	X-10 Transmit activity
BUSY	LynX-10 is busy processing requests
ERR	Error indicator
TW523	TW523 power is present when on
PWR	LynX-10 power is present when on

Back view of LynX-10 Coprocessor showing connectors



RS-232 Connector Pin Descriptions

PIN	DESCRIPTION
2	RX data to host from LynX-10
3	TX data from host to LynX-10
5	Ground
8	Clear to Send (CTS) from LynX-10 to host

Appendix B - Hexadecimal Numbering Overview

Table 5 - Hexadecimal and decimal relationship

Bit 3	Bit 2	Bit 1	Bit 0	Decimal	Hexadecimal
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

The HEXADECIMAL system is an outgrowth of the binary system of 1's and 0's used in computers. In hexadecimal, one character is used to define four bits. This is handy when using a large number of bits as in today's 32 bit microprocessors. Writing 32 ones and zeros in a row can take up some space. In hexadecimal (usually called "HEX"), it would only take 4. In the table below, all the possible combinations of the 1's and 0's are defined with both the decimal (known as base 10 - the numbers we humans use) and hexadecimal (base 16) numbers. Since our normal base 10 system only has 10 characters in it, we need to extend the characters with letters to reach 16. This is why A,B,C and so on, up to the letter F are used. This provides 6 more characters to indicate the remaining combinations. Putting two hex characters next to each other now provides a place holder for not 10 but 16. That is the most significant digit stands for the number of 16's there are. For example, if you had the hex value 32 (usually represented by 0x32 to indicate hex), there would be (3x16) + (2x1) counts or a decimal 50. The next digit would be the counts of 256 (16x larger), and the next would be the counts for 4096. If we had the hex value 1A5F, that would be equal to the decimal value of,

$$(1 \times 4096) + (10 \times 256) + (5 \times 16) + (15 \times 1) = 6751$$

or in binary: 0001101001011111. You can break the binary apart into 4 bit groups and see where the hex number is derived from. The 0001 is 1, 1010 is A, 0101 is 5, and 1111 is F. You can see how convenient this method is. There are calculators designed to do math in base 16. The Windows 3.1 Calculator program has a scientific mode that supports hex arithmetic. Try it out for your self. Hexadecimal numbers are sometime written with a subscript 16 like this: **19**₁₆ which indicates base 16. The 19 in this case would be equivalent to a decimal 25. If they are mixed with base 10 numbers, this notation really helps.

Most computer books discuss hexadecimal arithmetic, so if this is still not clear, you may wish to make a trek to your favorite book store and check out the ever growing computer section for some help.

Shown below in table 6 is the American Standard Code for Information Interchange or ASCII. The left axis is the least significant digit in both hex and binary, and the top is the most significant digit. This table is provided for your reference.

MSD LSD	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	NUL	DLE	SP	0	@	P	'	p
1 0001	SOH	DC1	!	1	A	Q	a	q
2 0010	STX	DC2	"	2	B	R	b	r
3 0011	ETX	DC3	#	3	C	S	c	s
4 0100	EOT	DC4	\$	4	D	T	d	t
5 0101	ENG	NAK	%	5	E	U	e	u
6 0110	ACK	SYN	&	6	F	V	f	v
7 0111	BEL	ETB	'	7	G	W	g	w
8 1000	BS	CAN	(8	H	X	h	x
9 1001	HT	EM)	9	I	Y	i	y
A 1010	LF	SUB	*	:	J	Z	j	z
B 1011	VT	ESC	+	;	K	[k	{
C 1100	FF	FS	,	<	L	\	l	
D 1101	CR	GS	-	=	M]	m	}
E 1110	SO	RS	.	>	N	↑	n	~
F 1111	SI	VS	/	?	O	←	o	DEL