LynX-10™ COPROCESSOR

Bi-directional RS-232 to X-10 Controller

Model 102A

Marrick Limited LynX-10™ Coprocessor
Manual revision 4.01
Marrick Limited, Incorporated
P.O. Box 950940
Lake Mary, FL 32795
(407) 323-4467 Voice
(407) 324-1291 FAX
EMAIL: techsupport@marrickltd.com
Web site: www.marrickltd.com

Table of Contents

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 “N0CE”. 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. OFF Command Summary

Command	Description	Example / Comments
F0*xy*	Turn off single unit *xy* (*xy* from table 1)	"F03F" turns off unit D16
F1*x*	Turn off all lights house code *x* (*x* from table 2)	"F12" turns off all lights on house code C
F2*x*	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

3.1.1 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.

3.2 ON Command Summary

Command	Description	Example / Comments
N0*xy*	Turn on single unit *xy (light or appliance)* (*xy* from table 1)	"N0F0" turns on unit P1
N1*x*	Turn on all lights this house code (*x* from table 2)	"N15" turns on all lights on house code F
N2	Turn on all lights, all house codes	"N2" turns on all lights on every house code
N3 - NF	(Reserved)	Returns error E1

3.2.1 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.

3.3. DIM Commands

3.3.1 DIM Command Summary (MODE 0)

Command	Description	Example / Comments
D*nxy*	Dims unit *xy* to level *n* (*xy* from table 1, *n* from table 3)	"DB00" Dims unit A1 to level 11

3.3.2 DIM Command Summary (MODE 1)

Command	Description	Example / Comments
D0*xn*	Sends *n* number of DIMs to house code *x* (*x* from table 2, *n* from table 4)	"D035" sends 6 DIMs to house code D
D1*xn*	Sends n number of BRIGHTs to house code x (*x* from table 2, *n* 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
S0*x*	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
S1*x*	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
S2*xy*	Requests status of unit *xy* (*xy* from table 1) ***note: see text***	"S2CA" requests current status for unit M11
S3	Returns internal status of coprocessor	"S3" returns S=xx where xx is the hex status byte
S4 – 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.

The S3 command returns the internal status of the LynX-10 Coprocessor. The assignment of the bits for the status byte are shown in the tabke below. The return value is made up of 2 ASCII characters representing the actual byte value of the status byte. For instance, after issuing an "S3", the LynX-10 may return "S=02". The "02" characters represent the binary value of 00000010 which indicates that the line frequency is 50 Hz.

BIT	Description
0	Lost power at TW523 / PSC05 (0=Power OK, 1=Lost power)
1	Power line frequency (0=60Hz, 1=50Hz)
2	EEPROM (0=Missing / not working, 1=Present and working)
3	XOFF Character (0=Not sent, 1=Sent)
4	XON Character (0=Not sent, 1=Sent)
5	(Reserved)
6	(Reserved)
7	(Reserved)

Table 5 – Status byte description

3.5. DIRECT X-10 ACCESS Command Summary

Command	Description	Example / Comments
X0*xy*	Addresses unit *xy* for next command (*xy* from table 1)	"X067" address X-10 unit G8
X1*x***0**	Sends an ALL UNITS OFF command to house code *x* (*x* from table 2)	"X1F0" sends ALL UNITS OFF to house code P
X1*x***1**	Sends an ALL LIGHTS ON command to house code *x* (*x* from table 2)	"X141" sends ALL LIGHTS ON to house code E
X1*x***2**	Sends an ON command to house code *x* (*x* from table 2)	"X102" sends ON to house code A
X1*x***3**	Sends an OFF command to house code *x* (*x* from table 2)	"X113" sends OFF to house code B
X1*x***4**	Sends a single DIM to house code *x* (*x* from table 2)	"X154" sends a DIM to house code F
X1*x***5**	Sends a single BRIGHT to house code *x* (*x* from table 2)	"X155" sends a BRIGHT to house code F
X1*x***6**	Sends ALL LIGHTS OFF command to house code x (x from table 2)	"X106" sends ALL LIGHTS OFF to house code A
X1*x***7**	Sends EXTENDED CODE to house code *x* (*x* from table 2) ***note: see text***	"X127" sends EXTENDED CODE to house code C
X1*x***8**	Sends HAIL REQUEST to house code *x* (*x* from table 2) ***note: see text***	"X1B8" sends HAIL REQ to house code L
X1*x***9**	Sends HAIL ACKNOWLEDGE to house code *x* (*x* from table 2) ***note: see text***	"X1B9" sends HAIL ACK from house code L
X1*n***A**	Sends PRESET DIM with MSB of 0 and *n* level (*n* from table 3) ***note: see text***	See application note AN-004 for details
X1*n***B**	Sends PRESET DIM with MSB of 1 and *n* level (*n* from table 3) ***note: see text***	See application note AN-004 for details
X1*x***C**	Sends EXTENDED DATA to house code *x* (*x* from table 2) ***note: see text***	"X10C" sends EXTENDED DATA to house code A
X1*x***D**	Sends STATUS ON for house code *x* (*x* from table 2) ***note: see text***	"X15D" sends STATUS ON to house code F
X1*x***E**	Sends STATUS OFF for house code *x* (*x* from table 2) ***note: see text***	"X15E" sends STATUS OFF to house code F
X1*x***F**	Sends STATUS REQUEST to house code *x* (*x* from table 2) ***note: see text***	"X12F" requests status from last addressed unit on house code C
X2 – XF	(Reserved)	Returns error E1

3.5.1 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 to 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.

3.6 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

3.6.1 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.

3.7 RESET Command Summary

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

3.8 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.

3.8.1 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".

3.9 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™ 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

3.10.1 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.

3.11 LynX-10™ REGISTER ACCESS Command Summary

Command	Description	Example / Comments
M00↵ or **M00=***xx*	MODE REGISTER 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**	SERIAL FIFO THRESHOLD 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**	X-10 RETRANSMISSION ATTEMPTS 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).
M03↵ or **M03=xx**	COMMAND RECEIVE TIMEOUT M03 followed by a carriage return will return to the host the current value of the TIMEOUT register. M03=xx will set the register to the desired timeout value in seconds. Setting this register to zero (M03=00) will disable this feature. Valid values of xx are 00-FF (0-255).	This function is used to prevent dead locking between a device and the LynX-10. If a command is not completely received by the end of timeout period, the LynX-10 is reset.
M04 through **MFF**	(Reserved)	Returns error E2

3.11.1 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" (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.***

3.12 LynX-10™ OUTPUT TO PORT Command Summary

Command	Description	Example / Comments
O*nn*=*xx*	Returns the value the of the output port on the LynX-10 device in the form O*nn*=*xx* where *nn* is the character string representing the HEX value of the port number and *xx* is the character string representing the HEX value of the port. **Port FF is reserved for the onboard LEDs (LED6-8). These LEDs are accessed via bits 0-2 respectively. Example: OFF=01 turns on LED6.**	**Note: The user must add additional circuitry in order to decode and use the I/O information from the processor.** See section 5 for more information.

3.13 LynX-10™ INPUT FROM PORT Command Summary

Command	Description	Example / Comments
I*nn*	Returns value of the input port *nn* on the LynX-10 device in the form Inn=*xx* where *xx* is character string representing the HEX value of the port pins and nn is the character string representing the HEX value of the port number.	**Note: The user must add additional circuitry in order to decode and use the I/O information from the processor**. See section 5 for more information

3.14 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). ***It also will clear ALL statistics counters to 0 and reset the run timer to 0.***	"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.

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

E2 : 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 AND X-10 RECEPTION

The Model 102 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***.

Received 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 / 98, UNIX, OS/2, and DOS.

5.3 POWER SUPPLY

The board is powered at J3 from a supply voltage of 12 to 25 volts AC or DC and consumes about 100mA of current. If expansion hardware is added, the regulator at U4 may need to be upgraded to an LM7805T (1 amp capable) and capacitor C12 increased to 100 uF or larger.

5.4 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.

5.5 ADDING CIRCUITRY FOR INPUT AND OUTPUT PORTS

This new version of the LynX-10 Coprocessor has hooks for adding circuitry to expand the functionality of the product. To accomplish this, the user should be intimately familiar with electronics and circuit design. Do not attempt modifying or adding circuitry unless you feel comfortable with a soldering station and schematics and have access to a logic analyzer or and oscilloscope. Shown below are the bus signals available from the processor on the prototyping area and the port timing diagrams. Check the Marrick Limited web site periodically to look for application notes and design ideas using additional circuitry.

5.5.1 Bus Control, Chip Selects, and Data signals on the Prototyping area

Shown in the following tables are the signal names, locations, and functions from the extended processor bus. The data bus is bi-directional and is pulled up with internal 20K resistors when idle.

Table 6 – Control Bus Signals

Location	Name	Use	Type
0	WRITE	Read / write control line for data bus (WRITE high)	OUTPUT
1	NC	(No connection)	N/A
2	INT0	Interrupt input to LynX-10 Coprocessor (Active low)	INPUT
3	INTA	Interrupt Acknowledge from LynX-10 C/P (Active high)	OUTPUT
4	LED100	LED 10 connection (Ground to turn on)	OUTPUT
5	LED90	LED 9 connection (Ground to turn on)	OUTPUT
6	RESET0	RESET to LynX-10 Coprocessor (Active low)	INPUT
7	NC	(No connection)	N/A

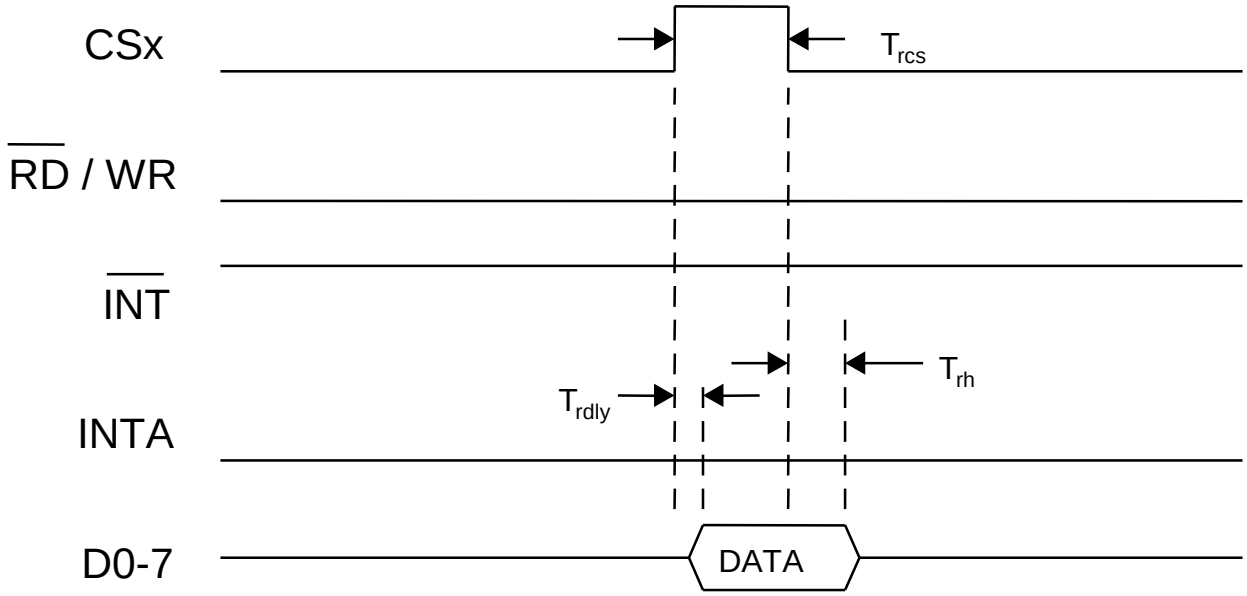
Table 7 – Chips Select Signals

Location	Name	Use	Type
0	CS0	Chip Select 0 – Decoded output for port 00 (active hi)	OUTPUT
1	CS1	Chip Select 1 – Decoded output for port 01 (active hi)	OUTPUT
2	CS2	Chip Select 2 – Decoded output for port 02 (active hi)	OUTPUT
3	CS3	Chip Select 3 – Decoded output for port 03 (active hi)	OUTPUT
4	CS4	Chip Select 4 – Decoded output for port 04 (active hi)	OUTPUT
5	CS5	Chip Select 5 – Decoded output for port 05 (active hi)	OUTPUT
6	CS6	Chip Select 6 – Decoded output for port 06 (active hi)	OUTPUT
7	CS7	Chip Select 7 – Decoded output for port 07 (active hi)	OUTPUT

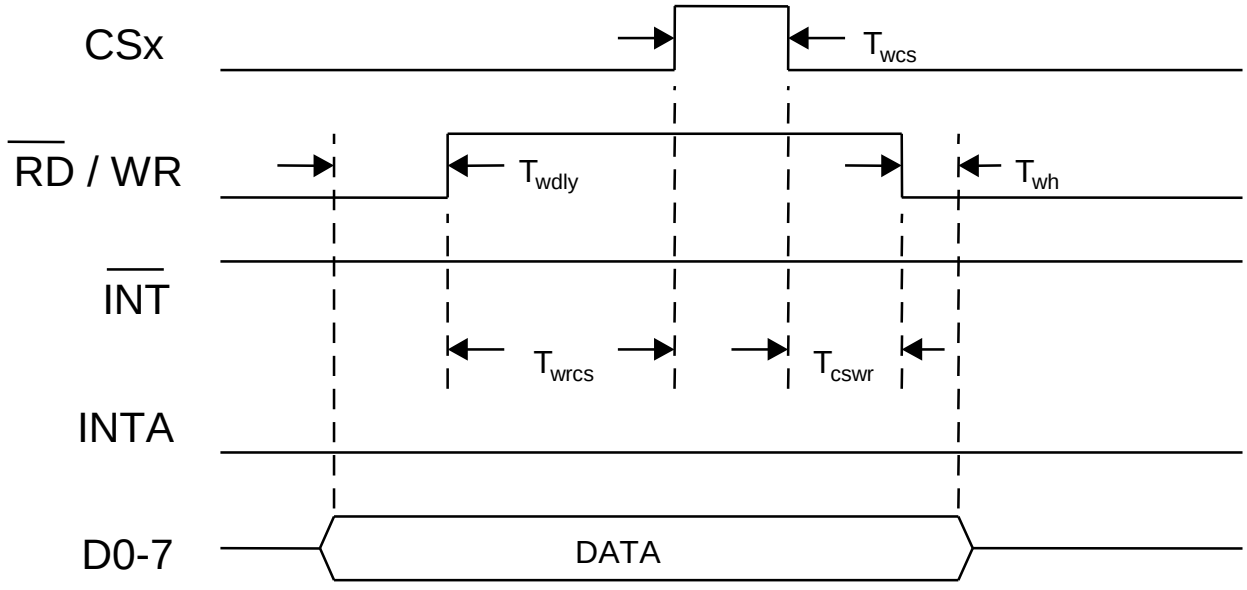
5.5.2 Timing Diagrams

Shown below are the timing diagrams for input and output operations to the optional ports. Table x shows the values of each timing symbols (i.e. Tread) and a description.

Timing Diagram 1 – I/O Port Read Timing



Timing Diagram 2 – I/O Port Write Timing



Timing Diagram 3 – I/O Port Interrupt Cycle Timing

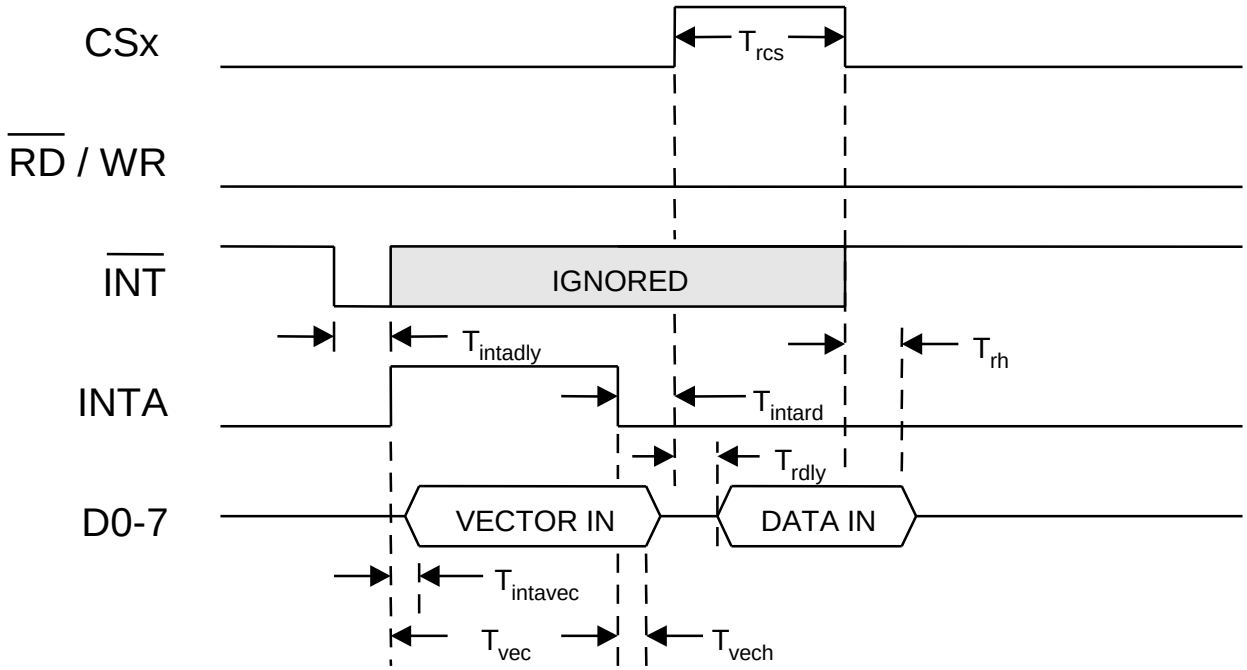


Table 8 - Timing Values

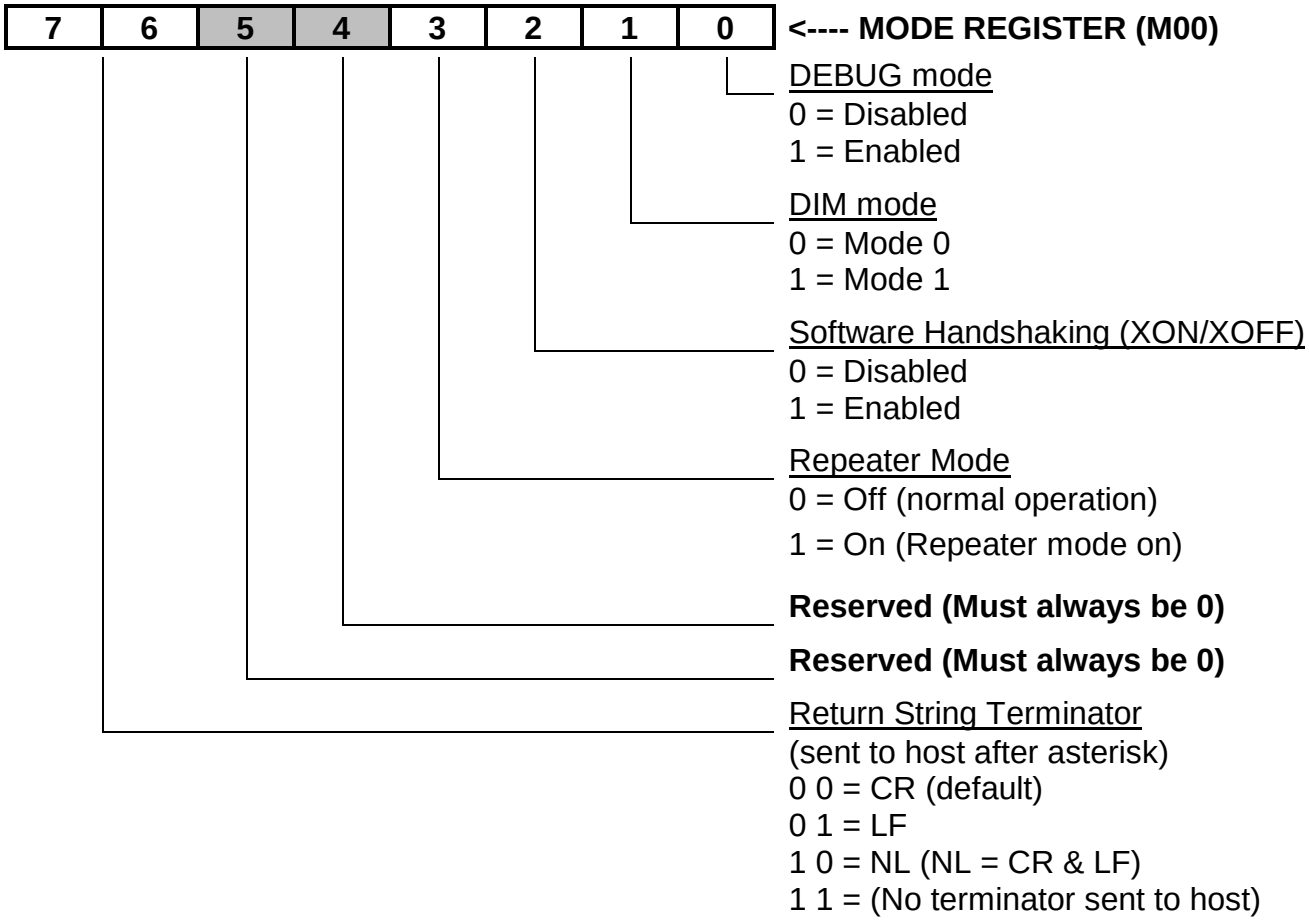
Symbol	Description	Min	Max	Units
Trcs	Read chip select length	6	7	uS
Trdly	Delay from Read to Valid Data	0	3	uS
Trh	Hold time for Read data after cycle ends	0	3	uS
Twcs	Write chip select length	3	4	uS
Tcswr	CS inactive to WRITE inactive	3	4	uS
Twrcs	WRITE active to CS active	6	7	uS
Twdly	Valid data to WRITE delay	3	4	uS
Tintadly	INT to INTA delay	0	20	mS
Tintard	INTA inactive to start of port read cycle	0	Note 1	uS
Tintavec	INTA to valid interrupt vector data	0	3	uS
Tvec	Length of Vector read	6	7	uS
Tvech	INTA inactive vector data hold	0	3	uS

Note 1: This delay could be indefinite if the LynX-10 is holding for completion of a command from the host. If not holding, delay should be less than 20 mS.

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 pre-decoded 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.

6.1.2 DIM Mode

The LynX-10 Coprocessor is capable of two methods of dimming. Mode 0 sets the absolute dim level of a light by first turning it off (sets to known state), and then turning it on and sending the correct number of DIM commands to set the requested level. Mode 1 is used for relative dimming. A computing device must keep track of the current level of all lights when using this mode. This mode allows the computer to send multiple DIM or BRIGHT commands without turning off the light. This looks much better than turning the light off first, however if anyone changes the state of a light without the computer knowing (i.e. turning off the light at the switch), synchronization will be lost with the light.

6.1.3 Repeater Mode

Repeater mode is new for firmware revisions 102-010 and higher. This mode allows two LynX-10 Coprocessors to be connected together with a Null modem cable and act as a repeater. When this bit is set in both LynX-10 Coprocessors, the two devices will communicate and repeat what each of them receives on the power line. Both units' TW523 must be isolated from each other for this option to work correctly. If they are not isolated, they will deadlock, repeating the same thing over and over and over… you get the idea.

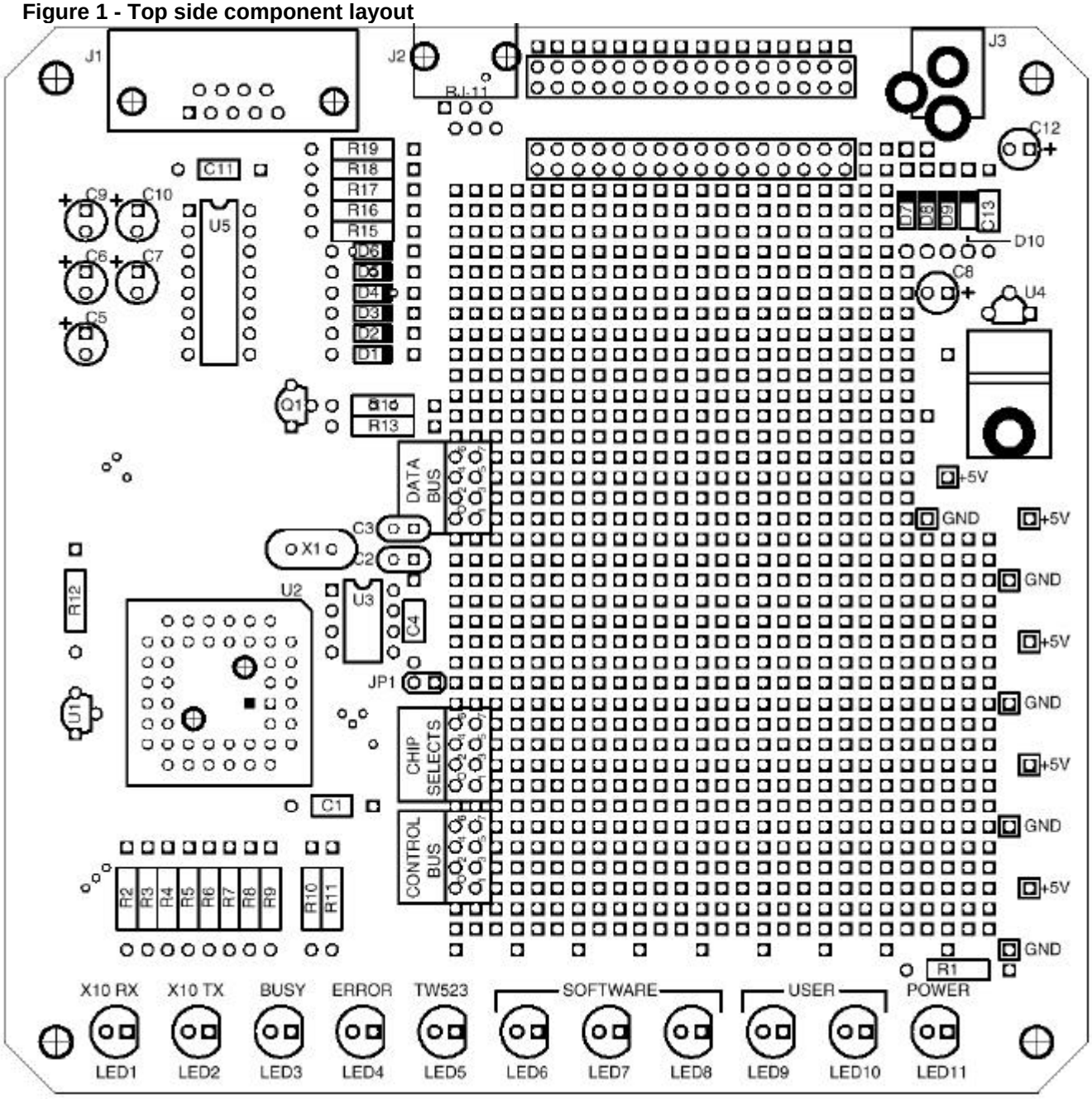
6.1.4 Return String Terminators

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.

Appendix A - BOARD LAYOUT NOTES

Shown below is the circuit board layout top (component) side (Marrick # 300-0102-0002-000). Please note that IC U4 can either be an LM7805T (TO-220 package) or an LM78L05Z (TO-92). These devices have different pin-outs. Please see the section below on the differences. When placing U4, pay attention to the silk screen on the PCB for orientation.

Figure 1 - Top side component layout



Differences between LM7805T (LM340T-5) and the LM78L05Z (LM340Z-5)

The "T" version package (TO-220) of the LM7805 is a large 3 lead package as shown in figure 2. The LM7805 is a 1 amp fixed 5 volt regulator, which requires this large package. This device has a little sister called the LM78L05. The LM78L05 comes in a smaller package (TO-92 transistor type) as shown in figure 3. Unfortunately, the two parts have exactly the opposite pin connections! This requires different insertion for each device. Since the board only draws less than 50mA of current, the LM78L05Z or LM340Z-5 (which are identical) are recommended. The kit includes either an LM78L05Z or an LM340Z-5. If other circuitry is used in the prototype area, and the total current will exceed 100mA, the LM7805T or LM340T-5 is required. Look closely at each figure to identify the correct pins for the component your using. This device will be inserted at location U4 (near the power connection). Make sure the input pin is the closest pin to the power connections. When assembling the board, install this IC before any of the other ICs. This will allow you to test the power supply to make sure you have +5V on the output of the regulator or on any of the supply pins of the other IC sockets. This will insure that you have not installed it backwards and potentially destroy the other integrated circuits... very bad form! :-)

Figure 2 - LM7805T (LM340T-5)

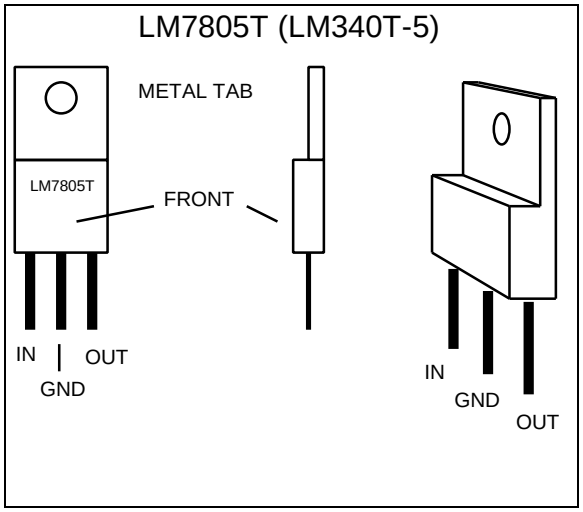
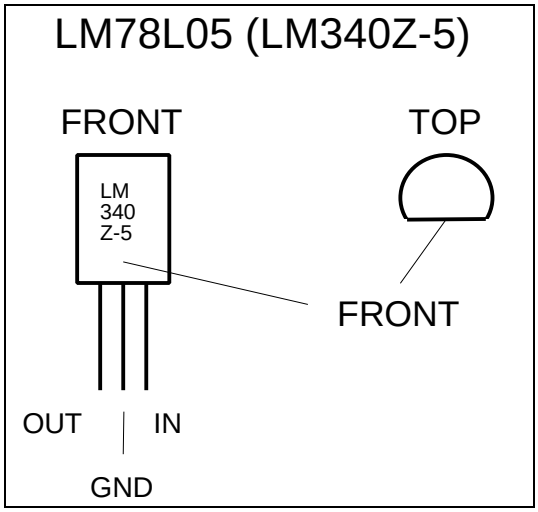


Figure 3 - LM78L05Z (LM340Z-5)



Appendix B - BILL OF MATERIALS

Qty	Description	Value	Location
1	PRINTED CIRCUIT	300-0102-002-000	N/A
4	CAPACITOR, CERAMIC	0.1 uF, 50V	C1, C4, C11, C13
2	CAPACITOR, CERAMIC	33 pF, 50V	C2, C3
4	CAPACITOR, ELEC.	4.7 uF, 16V	C6, C7, C9, C10
3	CAPACITOR, ELEC.	47 uF, 25V	C5, C8, C12
1	CONNECTOR, DB9	FEMALE 90 DEG	J1
1	CONNECTOR, RJ-11	6 POS. (PHONE)	J2
6	DIODE, SWITCHING	1N914	D1, D2, D3, D4, D5, D6
4	DIODE, RECTIFIER	1N4001	D7, D8, D9, D10
1	IC, EEPROM	NM93C06N	U3
1	IC, POWER MONITOR	DS1233-10	U1
1	IC, REGULATOR	LM78L05Z	U4 (Note: PCB will also accept LM7805T)
1	IC, RS232 TRANSCIEVER	DS14C232CN	U5
1	IC, CUSTOM MICRO	COP8SAC744V	U2 (Custom programmed by Marrick Limited)
2	LED, GREEN	T1-3/4	LED3, LED11
1	LED, RED	T1-3/4	LED4
8	LED, YELLOW	T1-3/4	LED1, LED2, LED5, LED6, LED7, LED8, LED9, LED10
1	OSCILATOR, CRYSTAL	10.000 MHz	X1
5	RESISTOR, CARBON	10K, 5%, 1/4W	R12, R13, R14, R15, R16
3	RESISTOR, CARBON	100, 5%, 1/4W	R17, R18, R19
11	RESISTOR, CARBON	330, 5%, 1/4W	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11
1	SOCKET, PLCC	44 PIN	U2
1	TRANSISTOR, NPN	PN2222A	Q1

Enclosures for the LynX-10 Coprocessor PCB

A standard enclosure for this board is available from Lansing Instrument Corporation from the address shown below. The PCB was designed for the MicroPak Q series, however, other series may also work. The stock length is 5 inches. Contact Lansing Instrument Corporation at 800-847-3535 for size, color selection, and pricing.

Lansing Instrument Corporation
PO Box 730
Ithaca, NY 14851
EMAIL: chaart@aol.com
800-847-3535

Appendix C - Hexadecimal Numbering Overview

Table 9 - 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,

(1x4096)+(10x256)+(5x16)+(15x1) = 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: $\mathbf{19_{16}}$ 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

