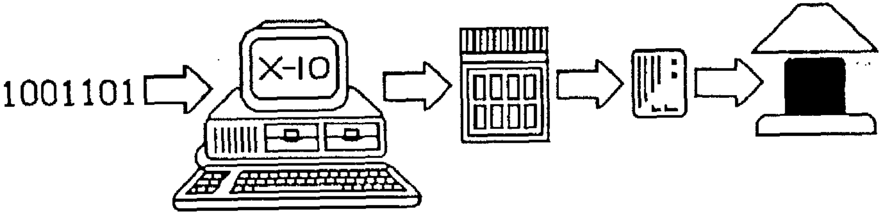
X-10® CP290 HOME CONTROL INTERFACE



PROGRAMMING GUIDE FOR ADVANCED PROGRAMMERS

Home Automation Systems, Inc. (800) SMART-HOME (949) 221-9200 http://smarthome.com

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

INTRODUCTION

Software is required to use your computer to program the X-10 Home Control Interface. The Interface is packaged with software and connecting cable for either the IBM PC (and compatibles), Apple Macintosh, Apple IIe/IIc or Commodore 64/128. That's all you need.

A Utility Program of Basic statements is included with the Home Control Software for IBM, Apple IIe/IIc and Commodore computers. These Utility programs let you write your own programs in Basic.

For more advanced programming you may also need to refer to this Programming Guide.

This Programming Guide is for advanced programmers who wish to write their own software using Machine Code and need more information than is supplied in the owner's manual which comes with the X-10 Home Control Software.

If you do NOT intend to write your own software, don't be intimidated by this Programming Guide - you don't need it.

The Interface must be connected to your computer for programming but once programmed it can be disconnected and will continue to send commands to the X-10 Modules under time control.

The main functions of the Interface are summarized as follows.

- Maintains a real time clock
- Stores the timed events relating to control of lights and appliances in the home.
- Stores the graphics data required by the computer to display the details of lamps and appliances installed into the house by the user.
- Transmits X-10 control signals onto existing house wiring to control lights and appliances connected to X-10 Modules.

A 9 volt Alkaline battery (sold separately) will provide approximately 100 hours back up for the Interface clock and stored data. When the Interface is running on battery power, the L.E.D. pulses approximately once every 5 seconds.

Up to 128 timer events + 256 ICONS (Graphical pictures of lights and appliances) can be stored in the Interface. A timer event is any number of unit codes on the same Housecode programmed to go on or off at a particular time at a specified brightness level on any day or days of the week. E.G. Modules Al, A4, A7 and Al5 programmed to go on at 70% brightness on Mondays, Wednesdays and Fridays at 7:30 p.m. is just one event and 128 events can be stored.

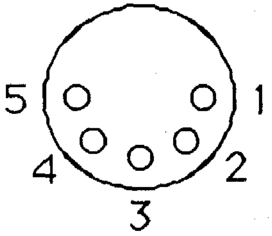
The Interface has 8 rocker keys to give manual control of unit codes 1 thru 8 on the Base Housecode. Base Housecode is set to “A” on power up but can be changed by software.

PROGRAMMING

The Interface is programmed to recognize 8 different types of instruction from the computer and each instruction has an ID number between 0 and 7. Each instruction from the computer has a leading SYNC pattern of 16 x FF bytes. The ID number tells the Interface what type of data to expect and a check sum is maintained which is compared with the last byte of data in the instruction. If the check sums agree, the Interface will acknowledge back to the computer and obey the instruction. It is suggested that if no response to an instruction is received within 10 seconds, the computer should advise the user of a potential problem with the connections to the Interface. E.G. the program could display a message such as “Error check Interface connections. Press Enter to continue”.

The Inter-face is programmed via the 5 pin DIN socket on the back of the Interface. The pin connections are as shown below.

LOOKING AT BACK OF INTERFACE



PIN	DESCRIPTION
1	-
2	**RECEIVE (INPUT)**
3	GROUND
4	**TRANSMIT (OUTPUT)**
5	-

The input signals from the computer (receive data input) are connected between pins 3 and 2.

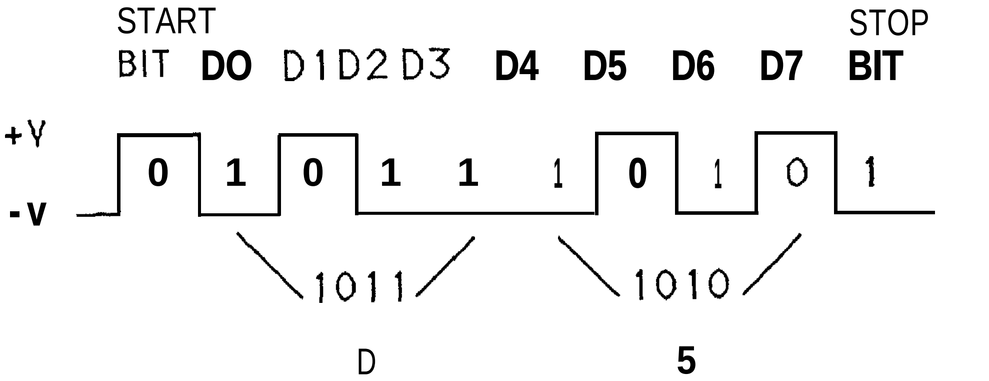
The output signals to the computer (transmit data output) are connected between pins 3 and 4.

A data cable is available for IBM, Macintosh, Apple IIe, Apple IIc and Commodore 64/128 and is included with the Interface and software for these computers.

Voltage levels meet RS-232 specifications and the data format is RS-232 with the following characteristics.

Baud rate : 600.
Data bits : 8.
Parity : None.
Stop bits : 1.

BYTE FORMAT



Dec □ 93 Hex = 5D 0 = Mark 1 = Space

A gap of 1 millisecond should be left between each byte of data sent.

A start bit signifies that a string of 8 data bits will follow. A start bit is always a SPACE bit, i.e. "0". A stop bit signifies that the data is finished and separates one byte from another. A stop bit is always a MARK bit, "1".

DOWNLOAD BASE HOUSECODE

When the Interface is first powered up, the Base Housecode is set to “A”. To change this you must first send a leading SYNC pattern of 76 x FF bytes to the Interface, followed by the identifier "0" for “download Base Housecode” and then a byte of data, the upper nibble of which contains the Housecode information. See below.

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
1-16	1	1	1	1	1	1	1	1	SYNC, 16 x FF.
17	0	0	0	0	0	0	0	0	ID 0, download Base Housecode.
18	HOUSECODE				0	0	0	0	See table 1.

TABLE 1

House Code	byte 18 (Hex)	House Code	byte 18 (Hex)	House Code	byte 18 (Hex)	House Code	byte 18 (Hex)
A	60	B	EO	C	20	D	A0
E	10	F	90	G	50	H	DO
I	70	J	FO	K	30	L	BO
M	00	N	80	O	40	P	co

Base Housecode is used by the rocker keys on the Interface. Changing the Base Housecode will reset all timer events and graphics data stored in the Interface, therefore before downloading a new Base Housecode to the Interface, the program should warn the user of this. E.G. the program could display a message such as 'Warning changing Base Housecode will erase all program information. Continue with. change yes/no".

After a successful download, the Interface will acknowledge by sending the “ACK” message to the computer.

ACK MESSAGE

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
I-6	1	1	1	1	1	1	1	1	FFX6
7	0	0	0	0	0	0	0	s	STATUS

The STATUS bit is reset to "0" during power up of the Interface and is set to “1” by a download of data from the computer (any data with byte 17 equal to ID 0, 1, 2, or 3). The STATUS bit is used to warn the computer that the Interface has been powered down. E.G. a STATUS bit equal to "0" could tell the program to display a message such as “The Interface has been powered down and contains no data. Press Enter to continue”.

DIRECT COMMAND (instant ON or OFF)

To turn something ON or OFF or adjust the brightness level of a light instantly, it is first necessary to send a leading SYNC pattern of 16 X FF bytes of data to the Interface. This is then followed by the identifier "1" for “direct command” and then 4 bytes of data followed by a check sum. The check sum is the sum of bytes 18 thru 21. See below.

BYTE	D7	D6	D5	D4	D3	D2	D1	D0	
1-16	1	1	1	1	1	1	1	1	SYNC FF X 16.
17	0	0	0	0	0	0	0	1	ID1, Direct command.
18	LEVEL				FUNCTION				See notes 1 and 2.
19	HOUSECODE				0	0	0	0	See table 1, page 10.
20	9	10	11	12	13	14	15	16	Bit mapped unit codes of
21	1	2	3	4	5	6	7	8	X-l 0 Modules.
22	CHECK SUM								Sum of bytes 18 - 21.

NOTE 1

“LEVEL” is dimmersetting for lamps. This applies only to X-l 0 Lamp Modules and Wall Switch Modules. Level F Hex is full DIM and level 0 Hex is full BRIGHT. The lamps specified by bytes 20 and 21 will switch on, adjust to full brightness and then DIM to the level specified by the upper nibble of byte 18. All codes between 0 Hex and F Hex are acceptable thus providing 16 discrete light levels.

NOTE 2

D3	D2	D1	DO	FUNCTION	EXPLANATION
0	0	1	0	ON	Modules with Housecodes as specified by upper nibble of byte 19 and unit codes as specified by bytes 20 and 21 will turn on.
0	0	1	1	OFF	As above, except turn OFF.
0	1	0	1	DIM	Lamp Modules and Wall Switch Modules addressed as above will turn on, adjust to full intensity and then DIM to the level specified by upper nibble of byte 18. Appliance Modules do not respond to bright and dim codes.

NOTE 3

If the check sum is accepted, the Interface will send the ACK response to the computer and will then transmit the X-10 codes onto the house wiring. When the power line transmission is complete the command is uploaded to the computer (see command upload, page 17). Also, if X-l 0 codes are transmitted by pressing the keys on the Interface, at the end of each transmission the codes are uploaded to the computer. This allows the computer to keep track of the ON/OFF status of the Modules while it is connected to the Interface.

DIRECT COMMAND EXAMPLES

Example 1: Turn ON modules A 1 and A4

BYTES	1-16	17	18	19	20	21	22
DATA	FF	01	02	60	00	90	F2

Example 2 : Turn OFF modules A 1 and A4

DATA	FF	01	03	60	00	90	F3

Example 3 : Turn ON lamp module B9 and DIM to 50%

DATA	FF	01	75	E0	80	00	D5

Example 4 : Turn OFF all modules with housecode A

DATA	FF	01	03	60	FF	FF	61

COMMAND UPLOAD (Interface to Computer)

This follows every transmission of X-10 codes onto the power line either from pressing the rocker keys, or from direct commands, or from timed events. This enables the computer to keep track of the ON/OFF status of lights and appliances.

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
I-6	1	1	1	1	1	1	1	1	FFX6.
7	0	0	0	0	0	0	0	s	Status.
8	HOUSECODE				FUNCTION				See note 4 below.
9	9	10	11	12	13	14	15	16	Bit mapped unit codes of
10	1	2	3	4	5	6	7	8	X-10 Modules.
11	BASE HOUSECODE				0	0	0	0	Same as table 1, page 10.
12	CHECK SUM								Sum of bytes 8-l 1.

NOTE 4 - Housecode same as table 1. Function same as note 2, page 4 except that the code for DIM is UPLOADED to the computer as 0100 (4 Hex).

SET CLOCK (Computer to Interface)

To set the clock in the Interface it is first necessary to send a leading SYNC pattern of 16 X FF bytes of data. This is followed by the identifier “2” for set clock and then 3 bytes of data followed by a check sum. This check sum is the sum of bytes 18 thru 20. See below.

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
1-16	1	1	1	1	1	1	1	1	SYNC 16 X FF.
17	0	0	0	0	0	0	1	0	ID2, for set clock.
18	0	0	MINUTES						HEX 00 to 3B (0 TO 59).
19	0	0	0	HOURS					HEX 00 to 17 (0 TO 23).
20	0	Sun	Sat	Fri	Thu	Wed	Tue	Mon	Bit mapped days.
21	CHECK SUM								Sum of bytes 18 to 20.

SET CLOCK EXAMPLES

EXAMPLE I To set clock to 9:30 a.m. **on Monday.**

BYTE	1-16	17	18	19	20	21
DATA	FF	02	1E	09	01	28

EXAMPLE 2 To set clock to 7:45 p.m. on Friday.

BYTE	1-16	17	18	19	20	21
DATA	FF	02	2D	13	10	50

TIMER EVENT OR GRAPHICS DATA DOWNLOAD

(Computer to Interface)

To download either a timer event or graphics data you must first send a leading sync pattern of 16 X FF bytes. The ID. (Byte 17) is 3 HEX for both timer events and graphics data but D2 in Byte 19 is a "0" for timer events and a "1" for graphics data.

Timer events are stored in bytes 0 to 1023 of the 2k x 8 RAM in the Interface. Only bytes 20 to 27 of the downloaded message are stored. Each event (group of 8 bytes) is assigned a Start Address in the RAM in the Interface. This Start Address is specified by AO-A4 in Byte 18 and A5-A6 in Byte 19. DO, D1 and D2 in byte 18 must always be 0, so that the Start Adresses increase in multiples of 8 (0, 8, 16, ... 1016). The computer should keep track of the event Start Adresses and load new events into vacant address locations in RAM. Byte 20 designates the type of timer event as shown in table 4. Bytes 21 thru 23 set the time and day of the event. Bytes 24 and 25 specify which Modules will be controlled and byte 26 specifies the Housecode of these Modules. Byte 27 specifies whether the Module(s) will turn ON, OFF or DIM and to what brightness level. Byte 28 is the sum of bytes 20 to 27. See page 21.

TIMER EVENT DOWNLOAD

BYTE	D7	D6	D5	D4	D3	D2	D1	D0	
1-16	1	1	1	1	1	1	1	1	SYNC 16 X FF
17	0	0	0	0	0	0	1	1	ID3, event/graphics download.
18	A4	A3	A2	AI	A0	0	0	0	A0 to A6, Binary coding of
19	x	x	x	x	x	0	A6	A5	event number.
20	0	0	0	0	MODE				See table 4, pages 22 &23.
21	0	Sun	Sat	Fti	Thu	Wed	Tue	Mon	Bit map of days.
22	0	0	HOUR						HEX 00 to 17 (0 to 23).
23	0	0	0	MINUTE					HEX 00 to 3B (0 to 59).
24	1	2	3	4	5	6	7	8	Bit map of unit codes.
25	9	10	11	12	13	14	15	16	Bit map of unit codes.
26	HOUSECODE				0	0	0	0	Same as table 1, page 10.
27	LEVEL				FUNCTION				Same as notes 1 and 2.
28	CHECK SUM								Sum of bytes 20 to 27.

X = DON’T CARE

TABLE 4 - TIMER MODE SELECTION

BYTE 20 lower nibble

D3	D2	D1	DO	MODE	EXPLANATION
1	0	0	0	NORMAL	occurs on a weekly cycle at same time each day, on day or days specified by byte 21 and at the time specified by bytes 22 and 23. The function and codes for the event are specified by bytes 24 to 27.
1	0	0	1	SECURITY	Same as NORMAL mode except that the event time will be different each day and will be within one hour after the time specified by byte 22. (varies in a pseudo random pattern). SECURITY is only available in EVERYDAY and SPECIFIC DAYS modes, see note 5, page 24.

TABLE 4 - TIMER MODE SELECTION (continued)

BYTE 20 lower nibble

D3	D2	D1	D0	MODE	DESCRIPTION
0	1	0	0	TODAY	EVENT occurs only TODAY at the time specified by bytes 22 and 23. and will be cleared from memory at midnight TODAY.
0	0	1	0	TOMORROW	EVENT occurs only TOMORROW at the time specified by bytes 22 and 23. and will be cleared from memory at midnight TOMORROW.
0	0	0	0	CLEAR	Clears from memory, the event specified by the event number stored in bytes 18 and 19.

NOTE 5

In addition to TODAY and TOMORROW, it is suggested that the program offer the user the choice of EVERYDAY and SPECIFIC DAYS. If EVERYDAY is chosen, byte 21 should be sent as 7F HEX (all days selected). If SPECIFIC DAYS is chosen, byte 21 should indicate which days were chosen.

GRAPHICS DATA DOWNLOAD

Graphics data is stored in bytes 1024 to 1535 of the 2k x 8 RAM in the Interface. Only bytes 20 and 21 of the downloaded message are stored. Each pair of bytes is assigned a number between 0 and 511 as specified by A0 to A6 in byte 18 and A7 in byte 19. DO in byte 18 is always '0' so these address numbers increase in steps of 2 (for graphics type and X-10 code of 256 objects). Note also that in byte 19 D1 is afways "0" and D2 is always "1". The computer should keep track of the message numbers and load new messages into vacant address locations. The contents of bytes 20 and 21 depends on the graphics approach used by the programmer (see note 6), the Interface merely stores this data and will upload it to the computer upon request (see graphics upload, page 33). Byte 22 is the sum of bytes 20 and 21. See page 25.

GRAPHICS DATA DOWNLOAD

BYTE	D7	D6	D5	D4	D3	D2	D1	D0	
I-16	1	1	1	1	1	1	1	1	SYNC 16 X FF
17	0	0	0	0	0	0	1	1	ID3, event/graphics download.
18	A6	A5	A4	A3	A2	Al	A0	0	A0 to A7, binary number for
19	X	X	X	X	X	1	0	A7	graphics object - 256 objects.
20	GRAPHICS DATA								User RAM to define type and
21	GRAPHICS DATA								X-10 code of graphics object.
22	CHECK SUM								Sum of bytes 20 and 21.

X = DON'T CARE

NOTE 6

A suggested allocation for byte 20 is shown below.

BYTE 20	D7	D6	D5	D4	D3	D2	D1	DO
	1=ON 0=OFF			ICON TYPE				

FOR EXAMPLE

ICON of a lamp shown in the ON state.	1	0	0	0	0	0	0	1
ICON of a T.V. shown in the ON state.	1	0 .	0	0	0	0	1	0
ICON of a coffee pot shown in the ON state.	1	0	0	0	0	0	1	1
ICON of a fan shown in the OFF state.	0	0	0	0	0	1	0	0

Byte 20 = 0 indicates a vacant ICON storage location. To clear an ICON from the Interface you need to send a graphics download with byte 20 = 0. Note, after doing this, you should also send a DOWNLOAD TIMER EVENT message with byte 20 = 0 (to clear any timed events for the removed ICON).

A suggested allocation for byte 21 is shown below.

BYTE 21	D7	D6	D5	D4	D3	D2	D1	DO
	HOUSECODE OF STORED ICON				**UNIT CODE OF STORED ICON**			

REQUEST CLOCK AND BASE HOUSECODE

(Interface to Computer)

To upload the time and Base Housecode from the Interface it is first necessary to send a leading SYNC pattern of 16 X FF bytes, followed by an ID4 for request clock and Base Housecode. See below.

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
1-16	1	1	1	1	1	1	1	1	SYNCFFX16
17	0	0	0	0	0	1	0	0	ID4, request clock and Base Housecode.

If the Interface receives the request correctly it will respond by uploading the clock and Base Housecode to the computer, as shown on page 29. If the request is not received correctly, no response is given.

CLOCK AND BASE HOUSECODE UPLOAD

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
1-6	1	1	1	1	1	1	1	1	SYNC 6 X FF.
7	0	0	0	0	0	0	0	s	status bit. •
8	0	0	MINUTES						HEX 00 to 3B (0 to 59).
9	0	0	0	HOURS					HEX 00 to 17 (0 to 23).
10	0	Sun	Sat	Fri	Thu	Wed	Tue	Mon	Bit mapped days.
11	BASE HOUSECODE				0	0	0	0	Same as table 1.
12	CHECK SUM								Sum of bytes 8 to 11.

* The STATUS bit is reset to "0" during power up of the Interface and is set to “1” by a DOWNLOAD of data from the computer (any data with byte 17 equal to ID 0, 1, 2, or 3). The STATUS bit is used to warn the computer that the Interface has been powered down. E.G. a STATUS bit equal to "0" could tell the program to display a message such as " The Interface has been powered down and contains no data. Press Enter to continue”.

REQUEST TIMER EVENTS (Interface to Computer)

To upload the timer events from the Interface it is first necessary to send a leading SYNC pattern of 16 X FF bytes, followed by an ID5, for request timer events. See below.

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
1-16	1	1	1	1	1	1	1	1	SYNCFFX16
17	0	0	0	0	0	1	0	1	ID5, request timer events.

The Interface will respond by uploading to the computer, all of the 128 events starting with number 1, as shown on page 31. A vacant event space is represented by a single FF byte, this shortens the time for the upload. The check sum does not include these FF bytes.

TIMER EVENTS UPLOAD

EXAMPLE WHERE ONLY FIRST TWO EVENTS ARE PROGRAMMED

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
I-6	1	1	1	1	1	1	1	1	SYNC FF X 6.
7	0	0	0	0	0	0	0	s	Status.
8-15	EVENT NUMBER 1 AS DOWNLOADED, 8 BYTES.								
16-23	EVENT NUMBER 2 AS DOWNLOADED, 8 BYTES.								
24-149	1	1	1	1	1	1	1	1	FF X 126 to indicate 126 vacant event spaces.
150	CHECK SUM								Sum of bytes 8 to 23 (FF bytes ignored).

REQUEST GRAPHICS DATA (Interface to Computer)

To upload graphics data from the Interface it is first necessary to send a leading SYNC pattern of 16 X FF bytes followed by an lD6, for request graphics data. See below.

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
1-16	1	1	1	1	1	1	1	1	SYNCFFX16
17	0	0	0	0	0	1	1	0	lD6, request graphics data.

The Inter-face will respond by uploading to the computer, all of the 256 ICONS starting with number 1, as shown on page 33. A vacant ICON space is represented by a single FF byte, this shortens the time for the upload. The check sum does not include these FF bytes.

GRAPHICS DATA UPLOAD

EXAMPLE WHERE ONLY 5 ICONS ARE PROGRAMMED

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
1-6	1	1	1	1	1	1	1	1	SYNC FF X 6.
7	0	0	0	0	0	0	0	s	Status,
8-9			ICON NUMBER 1						2 bytes.
10-l 1			ICON NUMBER 2						2 bytes.
12-13			ICON NUMBER 3						2 bytes.
14-15			ICON NUMBER 4						2 bytes.
16-17			ICON NUMBER 5						2 bytes.
18-268	1	1	1	1	1	1	1	1	FF x 251 to indicate 251 vacant ICON spaces.
269			CHECK SUM						Sum of bytes 8 to 17 (FF bytes ignored.

DIAGNOSTIC

The Interface has a self test diagnostic routine which is initiated by sending a leading SYNC pattern of 16 X FF bytes followed by an ID7. Upon receiving this instruction, the lntetface will run a self check on **it’s own** hardware and software (firmware). The output of the Interface (pins 3 and 4) will go low for 10 seconds as part of this test. If the check is o.k. the Interface will respond by sending the ACK with status "0". If a fault is diagnosed, the Interface will send ACK with status "1" or will not respond.

BYTE	D7	D6	D5	D4	D3	D2	D1	DO	
1-16	1	1	1	1	1	1	1	1	SYNCFFX16
17	0	0	0	0	0	1	1	1	ID7, initiate self test.