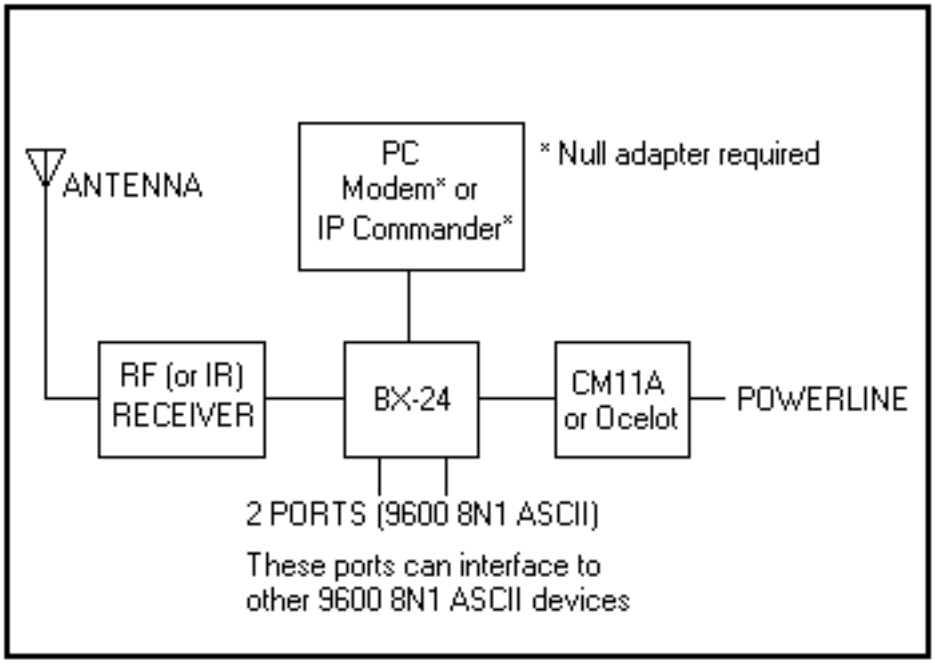
Contents

Overview

The BX24-AHT distribution package includes the BX-24 firmware, a companion Windows application that communicates with the BX24-AHT to configure it as well as send and receive commands and data when the BX24-AHT is not running standalone, a Windows application for downloading firmware to the BX24-AHT, initialization files for the Windows program and the downloader, and RFCodes.exe, a Windows program that generates codes for use with an RF-enabled OmniRemote™ Springboard (for the Handspring Visor) that can send X-10 and other RF codes. The installer puts all of these into the same folder.

This manual deals mostly with BX24-AHT.EXE, the Windows companion program, and with the OEM downloader program supplied by NetMedia, the company that manufactures the BasicX-24 microcontroller (hereafter referred to as BX-24). There will be versions of the Windows interface and firmware specific to the CM11A, IP Commander, Ocelot and more. This manual deals primarily with the CM11A version. There are appendices that cover features of the other versions.

The BX24-AHT is a DIY X-10 RF All Housecode Transceiver. Unlike standard X-10 RF transceivers, the BX24-AHT can handle all housecodes and, in addition to standard X-10 RF, can receive and decode RF commands sent by X-10 security devices, and Stanley Homelink garage door sensors. It also can receive proprietary RF commands sent by user built RF transmitter nodes for ADC and digital inputs or from an RF-enabled OmniRemote™ Springboard in a Handspring Visor PDA. It can even be configured for IR input in environments where RF is undesirable. In addition to an RS232 port for connection to a controlling PC, modem, or IP Commander, there are 4 serial ports, one of which is intended for a CM11A or Ocelot, one of which will be used for an RS485 network, and two that are configured for 9600 8N1 which can be used for ASCII output to other devices.

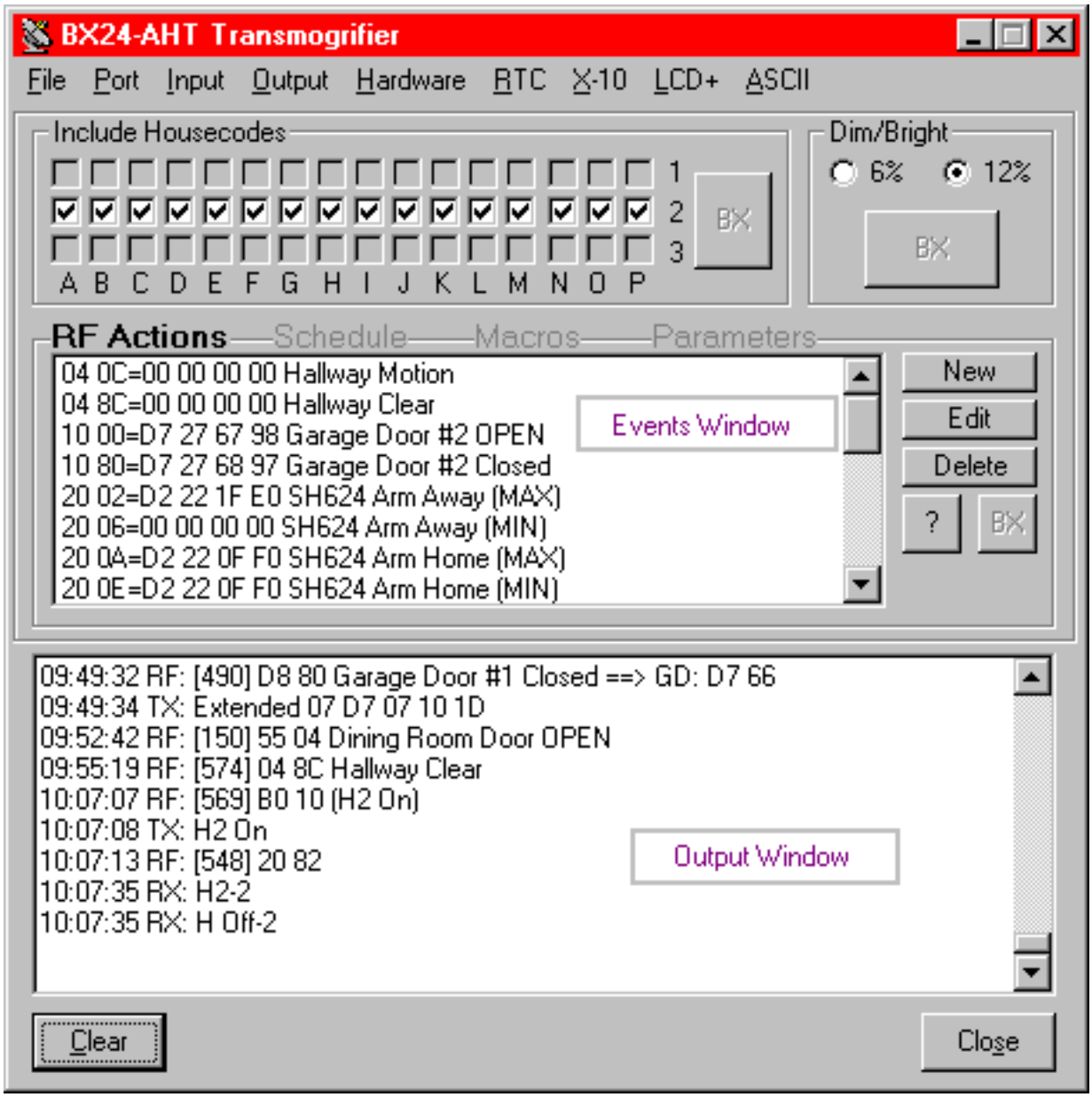


Since it is based on the BX-24, the BX24-AHT is readily and easily reconfigured merely by downloading new firmware using the same RS232 port that is used for the Windows interface program so it is unlikely to ever become obsolete.

Windows Interface

The main screen of the BX24-AHT Windows interface program is shown below. The upper portion of the screen is used for setting configuration details while the **Output Window** in the lower part of the screen reports all BX24-AHT activity to the PC RS-232 port. In Standard output mode, all X-10 powerline activity is reported whether or not the housecode is selected for X-10 RF reception and sending X-10.

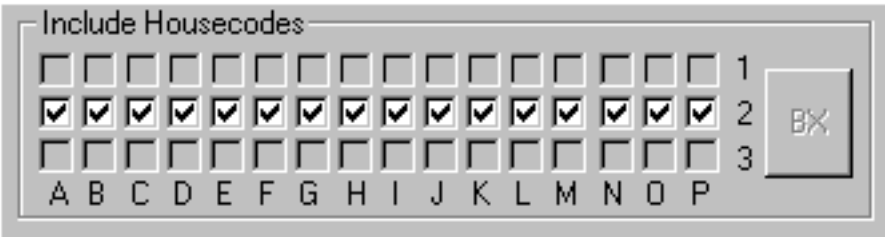
Once configured, the BX24-AHT can be disconnected from the PC and it will continue to run independently.



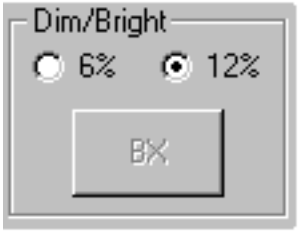
Configuration and event details are stored in the BX-24's non-volatile EEPROM which is rated for a finite number of erase/write cycles. While its useful life may be higher, NetMedia only guarantees the BX-24 EEPROM for 100,000 erase/write cycles. To prevent excessive erase/write cycles, the **BX** buttons, which write configuration data to EEPROM, are disabled after any EEPROM write cycle and remain disabled until new changes are made to the data.

EEPROM space is reserved for 100 **RF Action**s and for 47 **Parameters**. Since they do not have a constant size, the total number of **Schedule** events and **Macros** will vary but 2K of EEPROM is reserved for them.

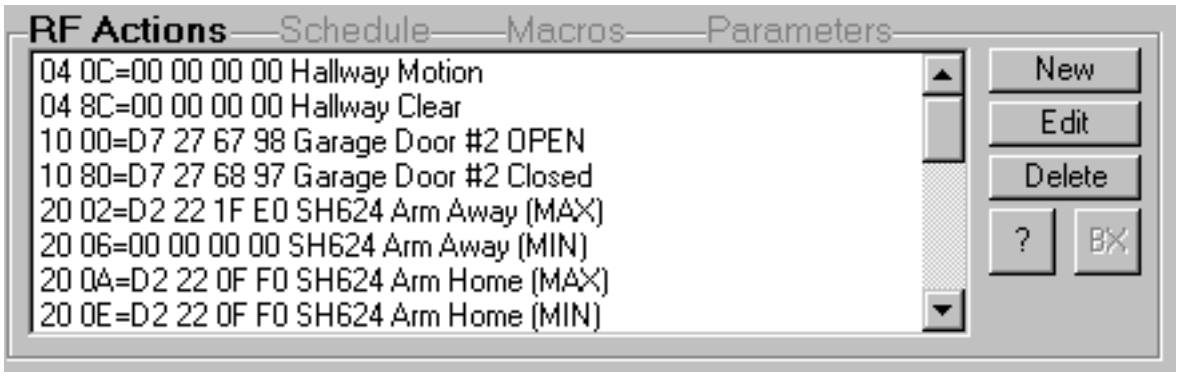
The **Include Housecodes** panel contains an array of check boxes that let you select those housecodes for which you wish the BX24-AHT to respond to RF commands. The row that is enabled will depend on the port that you select for the CM11A in the Hardware configuration menu. Future versions may allow for multiple powerline interfaces on separate electrical phases with housecodes segregated by phase. Note that the housecode must be checked both to receive RF and to send X-10 commands.



The **Dim/Bright** panel contains radio buttons for setting the default Dim/Bright level when the BX24-AHT responds to received RF Dim/Bright commands.



The **Events Window** is used for managing event definitions for the four types of events. **RF Actions** define the BX24-AHT response to received RF codes, **Schedules** define X-10 commands to be sent depending on time of day, **Macros** define X-10 commands to be sent in response to X-10 powerline receptions, and **Parameters** define ADC and digital inputs that the BX24-AHT will monitor automatically at user defined intervals.



Other configuration details are set from the Main Menu.

Main Menu

File Port Input Output Hardware RTC X-10 LCD+ ASCII

The **LCD+** and **ASCII** menu items will only be visible if these items are selected in the hardware configuration.

File Menu

BX-24 Loader...	launches the firmware downloader
Reset BX-24	restarts the BX-24 processor
Exit	closes the interface program

Port Menu

✔ COM1
COM2
COM3
COM4
COM5
COM6
COM7
COM8
COM9
COM10
COM11
COM12
COM13
COM14
COM15
COM16

Selects the PC COM port to which the BX24-AHT (DB-9) is connected. The Windows interface program can use COM1-16 but the NetMedia downloader can only use COM1-4 so you will need to use 1-4 until you have downloaded the firmware. NetMedia has another beta version of the downloader which can use COM1-16 which we will switch to once we are sure it is stable.

Input Menu

IR
✔ RF

Selects the wireless input mode.

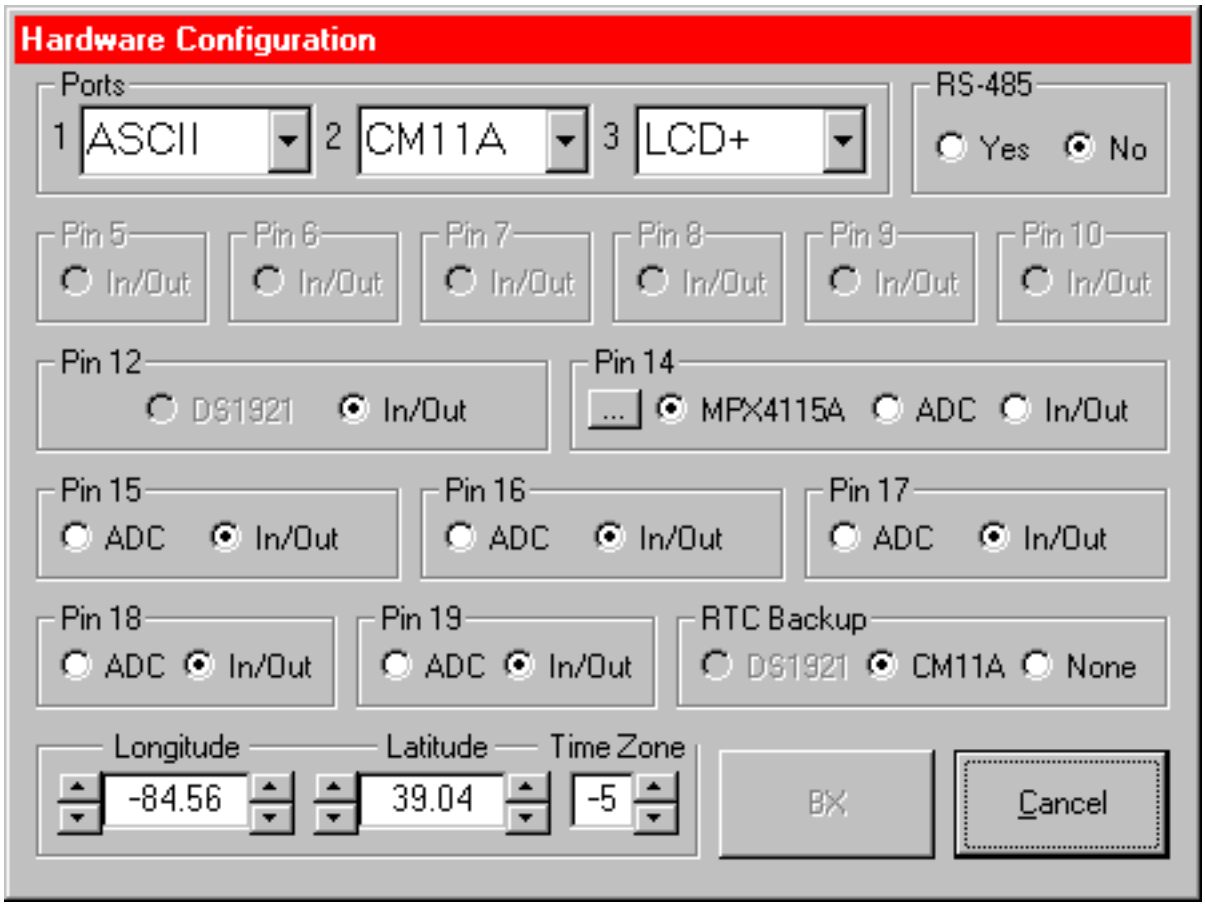
Output Menu

PC Only
✔ Standard
✔ Log to file

PC Only mode receives and decodes RF but only simulates X-10 output to the powerline and does not receive from the powerline. Its primary use is for trouble shooting and for familiarizing yourself with the features of the BX24-AHT. **Standard** mode is the normal operating mode. It outputs X-10 to the powerline using the CM11A. **Log to file** logs all activity to BX24-AHT.LOG which is in the same folder as the main program files. If you leave logging active, you should delete the log file periodically or it will get very large and waste disk space.

Hardware Menu - allows setting the hardware configuration.

See the **Hardware Configuration** section for details.



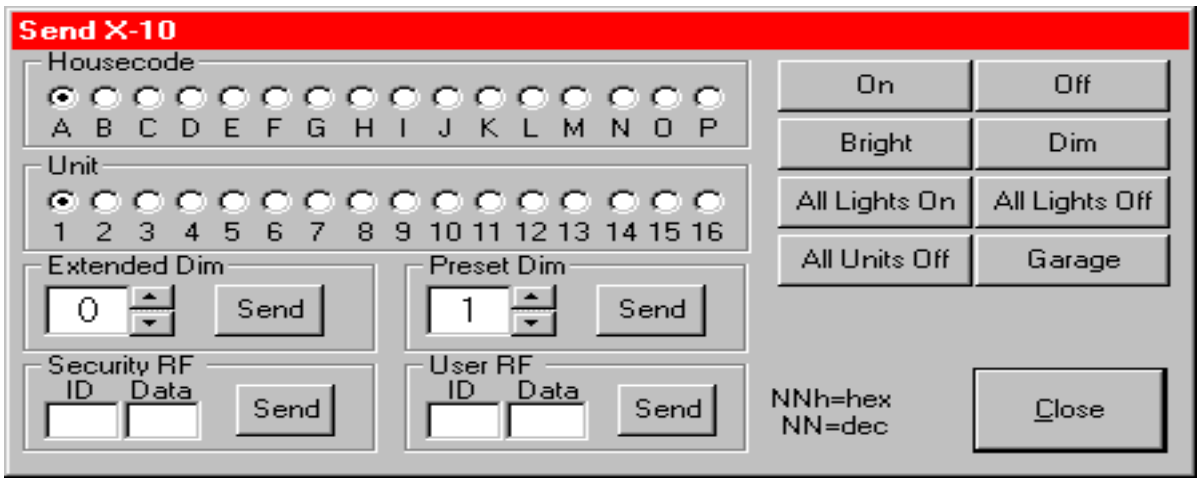
RTC Menu

The BX-24 includes a software RTC but it is not battery backed and resets on any power cycle or watchdog reset. The CM11A includes an RTC that is battery backed (if batteries are installed) that maintains accurate time. The BX24-AHT uses the CM11A RTC to sync the BX-24 RTC and then uses the BX-24 RTC for its time and date tracking. The BX24-AHT will retain the date through a power outage or watchdog reset unless the outage spans midnight. The BX-24 RTC is automatically synced whenever the CM11A RTC is set or read. Whenever new firmware runs for the first time, it may be necessary to reset the BX-24 RTC time and date.

When active, **Parameter** 2 will automatically sync the BX-24 RTC to the CM11A RTC every 239 minutes to correct BX-24 RTC drift.

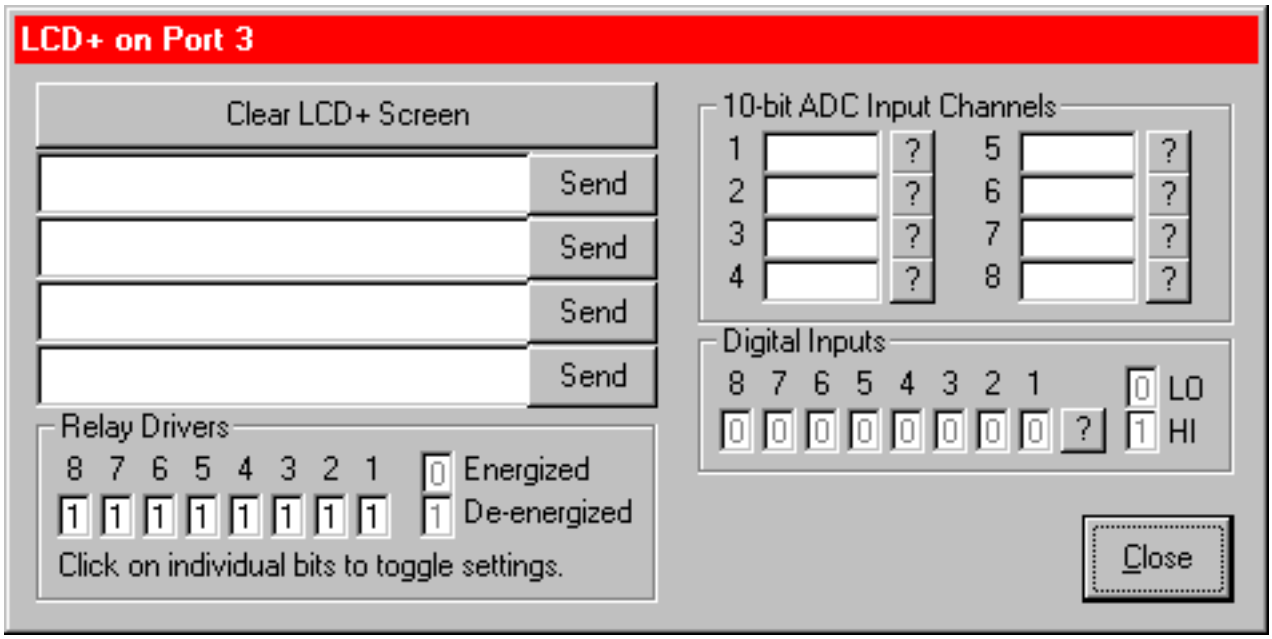
X-10 Menu - allows sending direct X-10 commands.

See the **Direct X-10 Commands** section for details.



LCD+ Menu - allows sending directly to LCD+

See the **Direct LCD+ Commands** section for details.



ASCII Menu - allows sending directly to ASCII devices

See the **Direct ASCII Output** section for details.

NetMedia Firmware Downloader

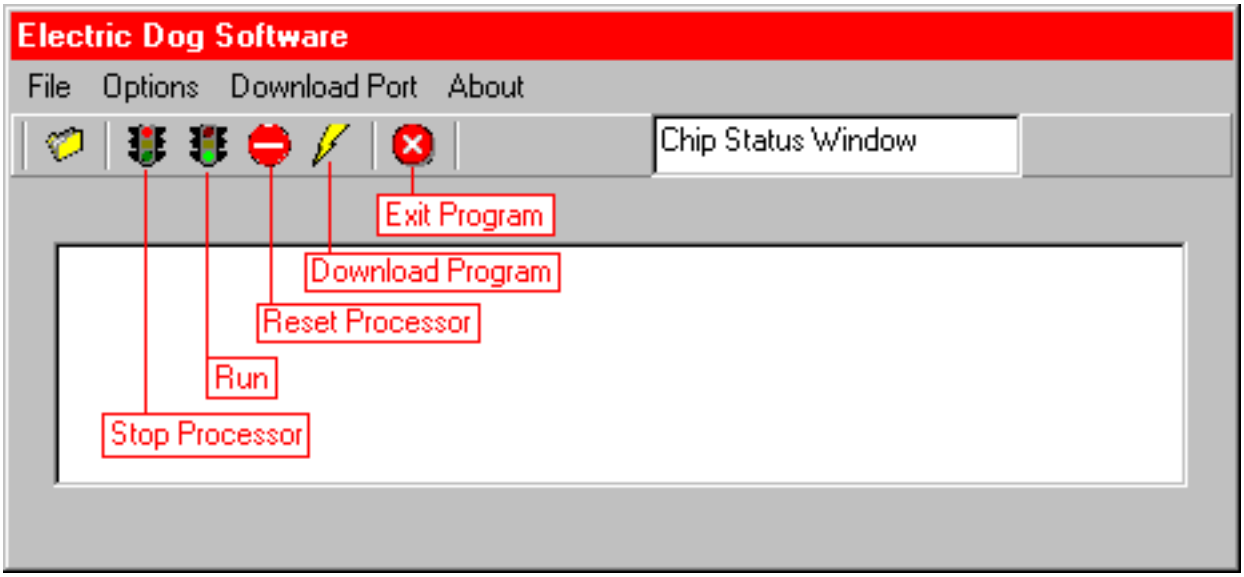
The BX-24 is shipped from the factory preloaded with a simple application that alternately flashes the red and green LEDs. You will need to replace this with the BX24-AHT firmware.

The BX24-AHT installer will put the firmware file in the folder where you installed the main program. The firmware will always have a filename beginning with **AHT** and a file extension of **.BXB**. Exact filenames will vary with the exact version but the first public release of the firmware is **aht30a.bxb**.

Clicking the "**File | BX-24 Loader...**" menu entry calls the OEM Downloader program supplied by NetMedia who manufactures the BasicX-24. The downloader window is shown below with labels for the buttons on the toolbar.

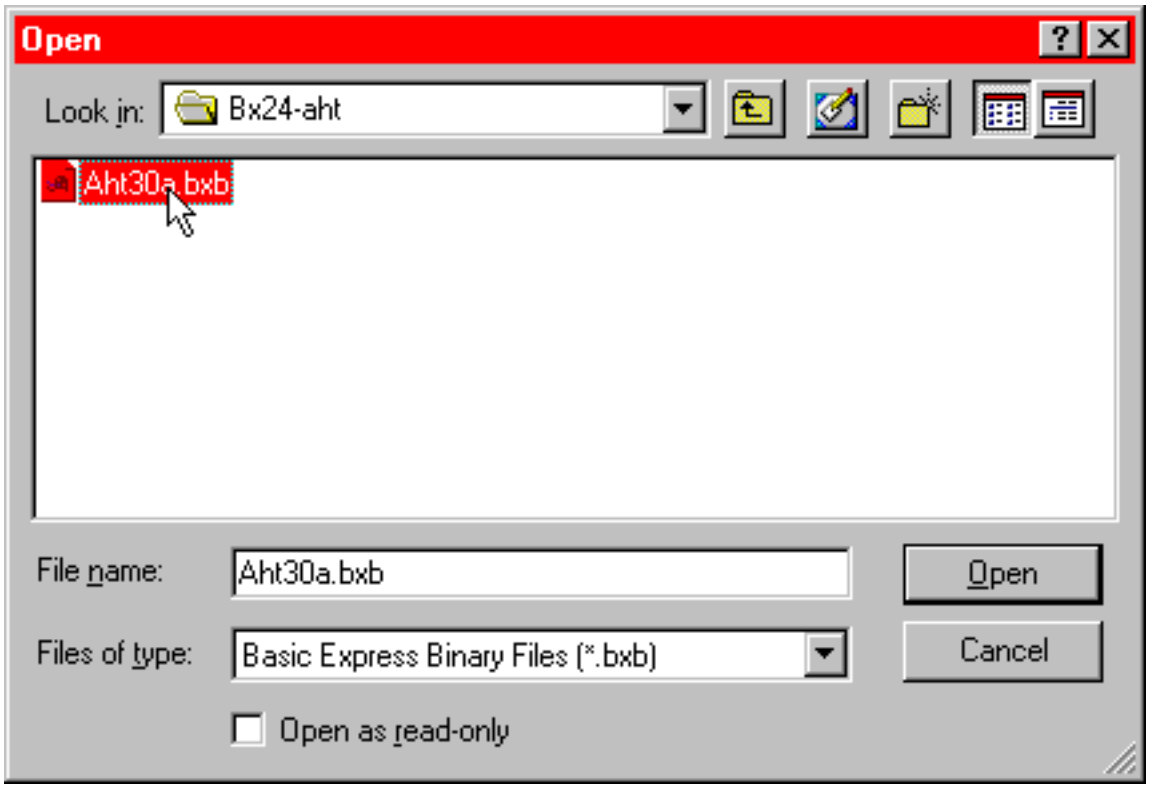
The OEM downloader is not the most polished and user friendly of programs but it does get the job done. We suggest you read through this section completely to get a preview of problems you may encounter.

One problem is that while the BX24-AHT Configuration program can use COM1-COM16, the downloader can only use COM1-COM4 so you will need to set both programs to use the same COM port. Unfortunately, NetMedia's programmers have not provided a way to do this automatically so you will have to select the "Download Port" from the downloader menu.

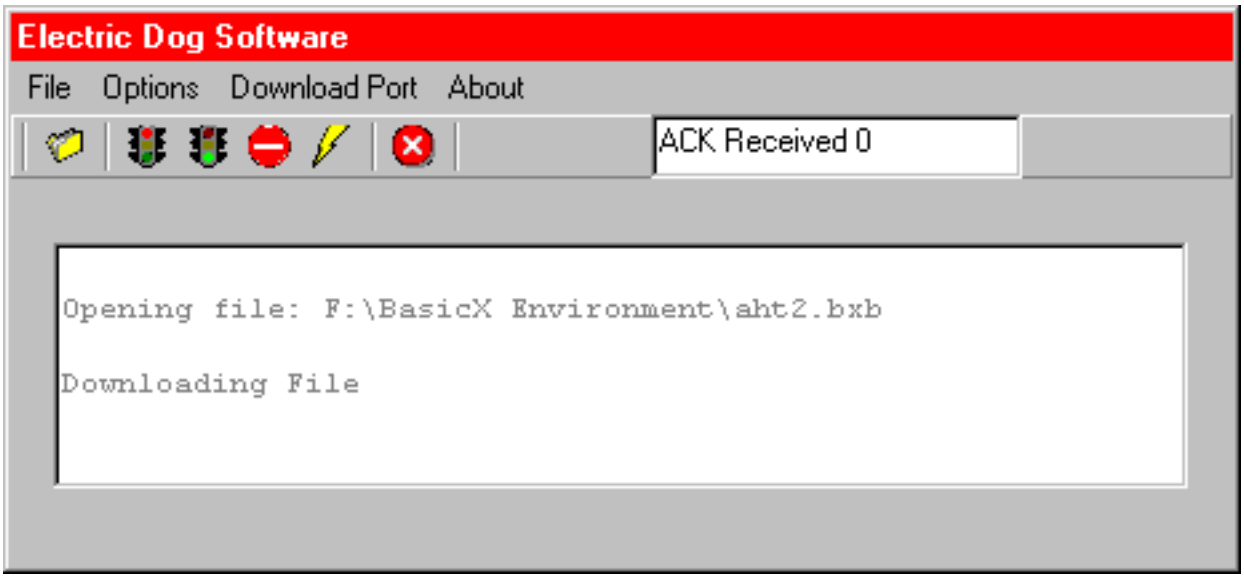


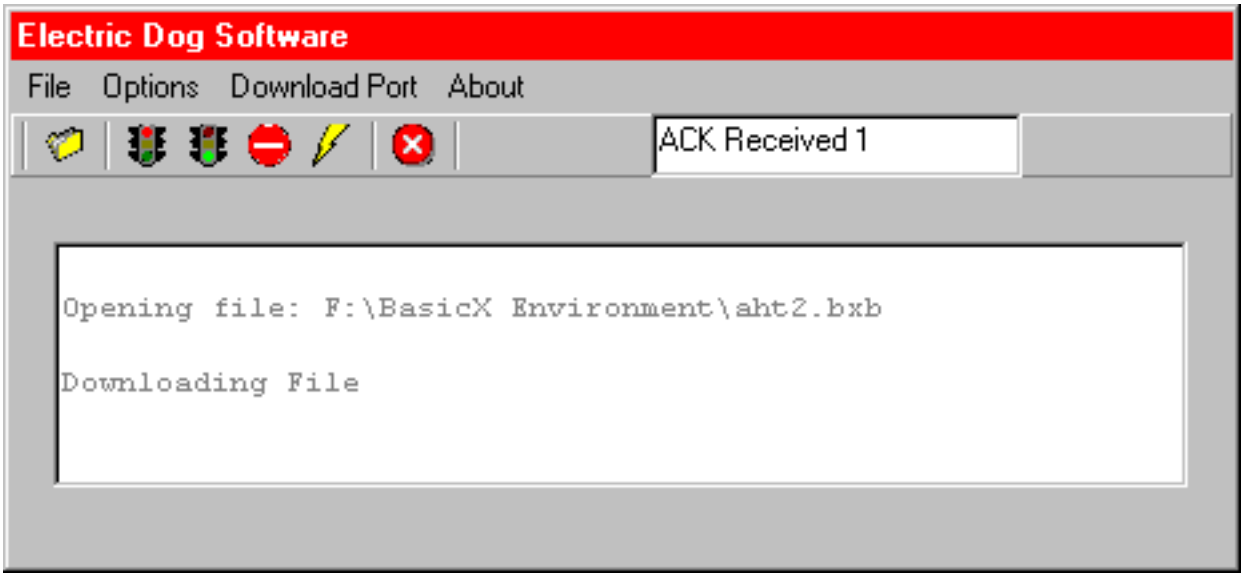
Follow these steps to download new firmware to the BX24-AHT.

1. Select the Download Port from the menu.
2. Click the **Folder** button and select the firmware file.
3. Click the **Stop Processor** button.
4. Click the **Download Program** button.



The OEM Downloader program will take control of the COM port, set it to its maximum speed, and download the .bxb file using SPI. If all goes well, the Chip Status Window will display some **ACK Received n** messages like the screens shown below and on the following page.





And then, after a a few seconds, you will see the screen below.

After a successful download:

1. Click the **Run** button to start the BX24-AHT firmware.
2. Click the **Exit Program** button to return to the BX24-AHT Configuration program.

IMPORTANT: You must click the **Exit Program** within a second or two of clicking the **Run** button. If you do, you will see the BX24-AHT copyright and sign-on messages in the lower window of the Configuration program and will be prompted to check and/or reset the various configuration parameters.

If too much time passes between clicking **Run** and **Exit Program**, the Windows configuration program will miss signals from the BX24-AHT and be out of sync with the firmware. If this happens, you will need to make changes to parameters that will enable the **BX** buttons and allow you to reset the BX24-AHT parameters.

If you close the Downloader without clicking the **Run** button, you can restart the processor using the hardware reset switch on the circuit board.

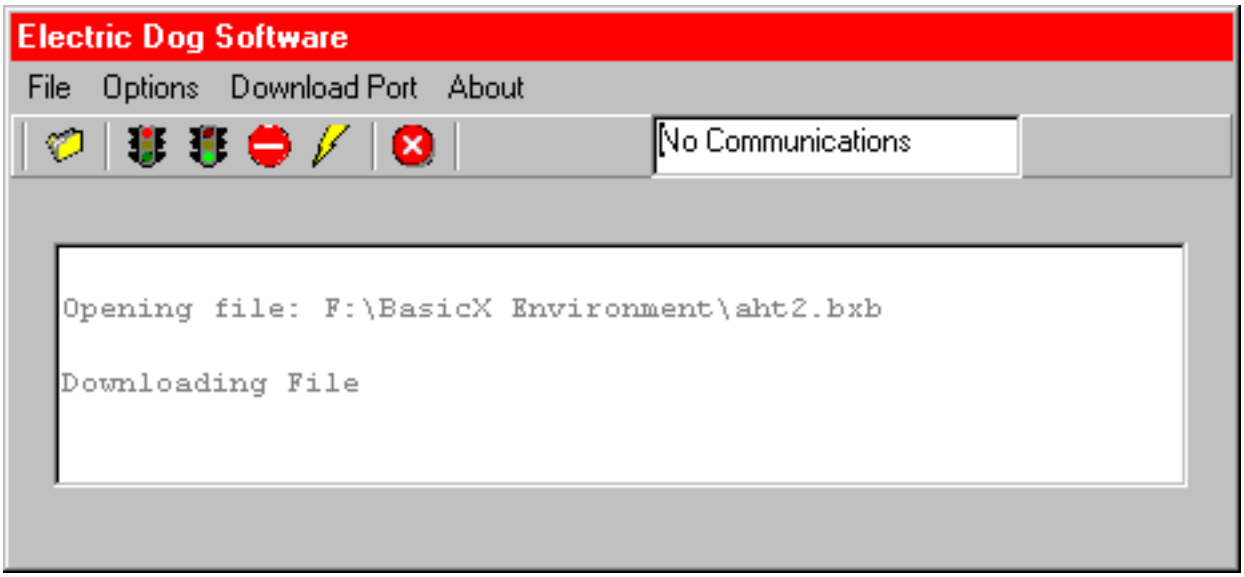
Whenever a new version of the BX-24 firmware is downloaded, the BX24-AHT defaults to:

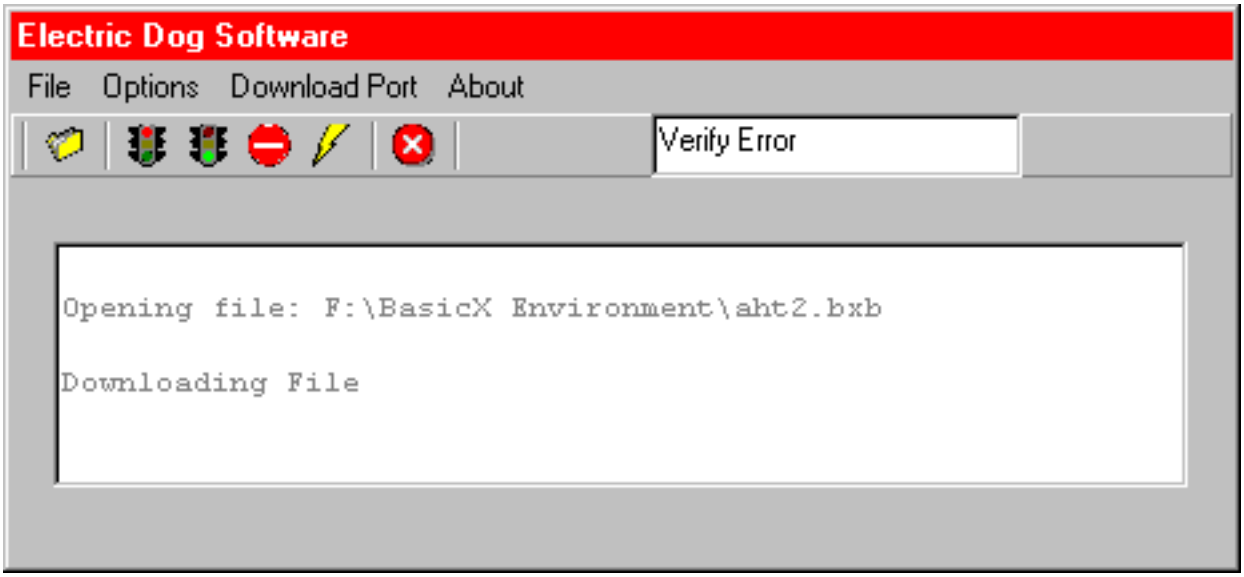
- RF input mode
- PC output mode
- Include all housecodes
- 6% Dim/Bright increment

And it will not know how the pins are configured for various hardware options. You will be prompted to reset the Dim/Bright default, Housecodes, Output mode, Hardware and the RTC as well as any Events that have been defined..

Thereafter, if you reset the BX-24, if it gets reset by the watchdog, or if it loses and regains power, it will retain all previous settings when it restarts.

The bad news is that sometimes all does not go well. In these cases you will see error screens like the two that follow.





The good news is that there are ways to recover from the errors.

The first step is to start Windows Task Manager and close the **BasicX** application.

Step two is to try downloading again by clicking the **Download Program** button.

If you still receive error messages, you will need to use the **Rescue** function from the **Download Port** menu. After clicking the Rescue menu item, you will be prompted to press the hardware reset button which is the Reset switch on the BX24-AHT circuit board.

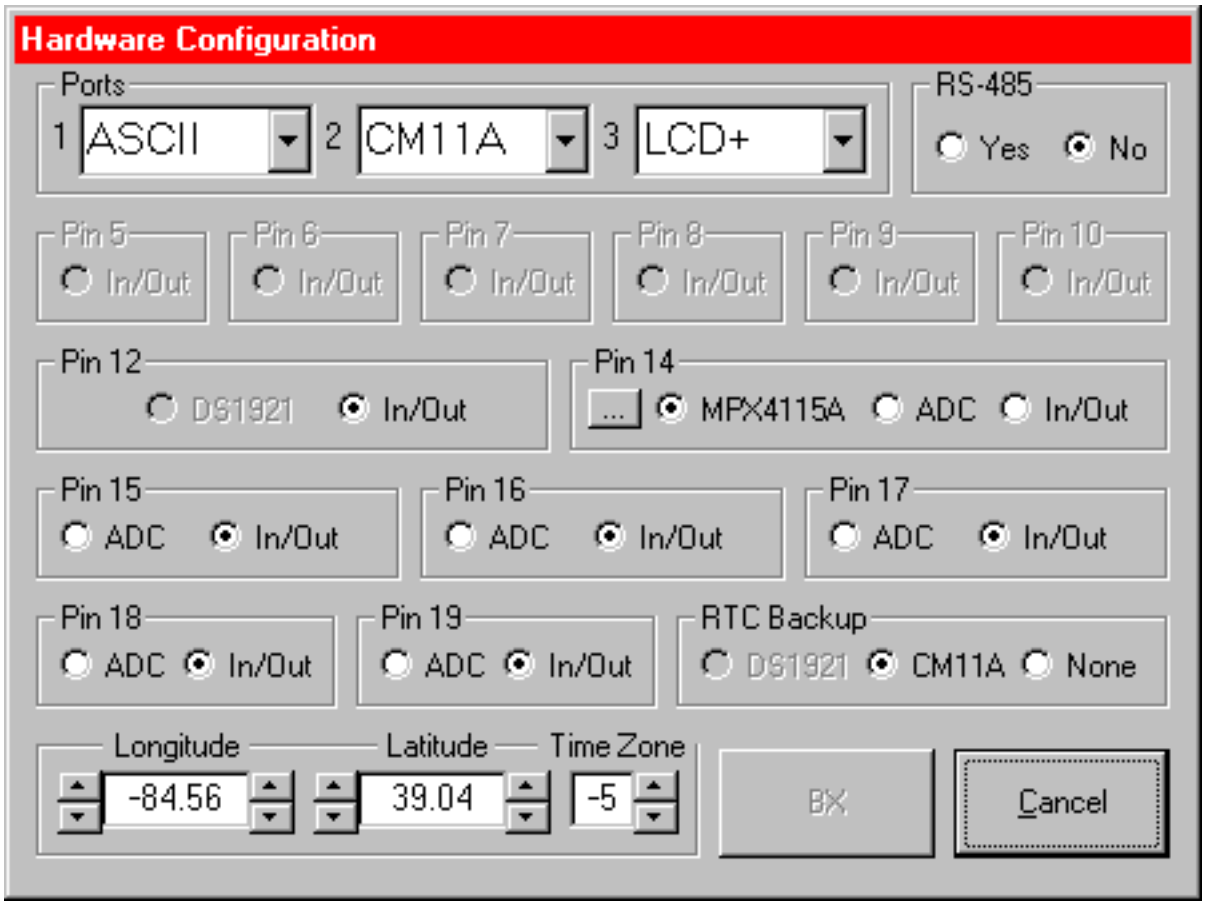
Then follow the prompts from the downloader.

And you should end up here.

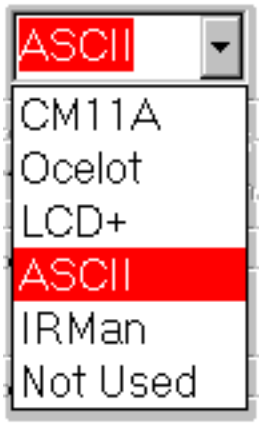
Please refer all complaints to NetMedia. They wrote the downloader.

Hardware Configuration

The BX24-AHT has been designed to maximize hardware flexibility. Most of the pins on the BX-24 MCU can be used for multiple purposes depending on the specific hardware installed and/or in use. (See the Construction Instruction manual on the web site for details on specific pins.) To configure the hardware, click the Hardware menu.



Other than Pins 18 and 19 which are always free for ADC or digital I/O, the options available in the **Hardware Configuration** panel are determined by the physical hardware installed and by the port assignments made for serial output ports 1-3 and the RS-485 port. The settings made in the Hardware configuration panel will determine the options that are available for the Parameters configuration.



Each RS-232 output port can be assigned one of the devices illustrated at left.

Only one CM11A and one LCD+ are supported.

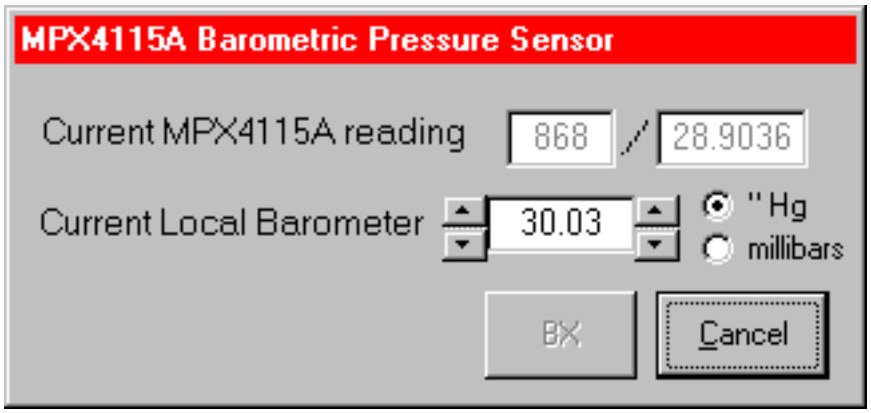
The output ports expect a DCE device. A null adaptor is required for a DTE device.

If IRMan is selected, the BX24-AHT will relay all RF input to the IRMan port emulating an IRMan. Use a null adapter to connect this port to the PC running the software that expects IRMan input.

Digital I/O pins can be set as Input or Output either in **Parameter** definitions or **RF Action** definitions. They are set or read on-the-fly. Pins 5-10 are only capable of being digital inputs or outputs. If they are not used for RS-232 output ports, the radio buttons are enabled merely as an indication that they are available for digital I/O.

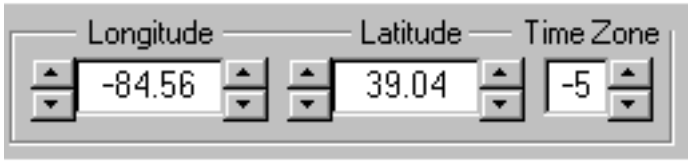
Barometric Pressure

The MPX4115AS is a pressure transducer for barometric pressure. To calibrate it to local barometric pressure, click the button.



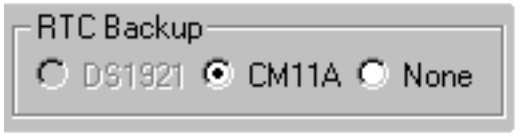
Select units, adjust the spin controls to agree with local weather service readings, and click the **BX** button to save the k-factor to the BX24-AHT EEPROM.

Location



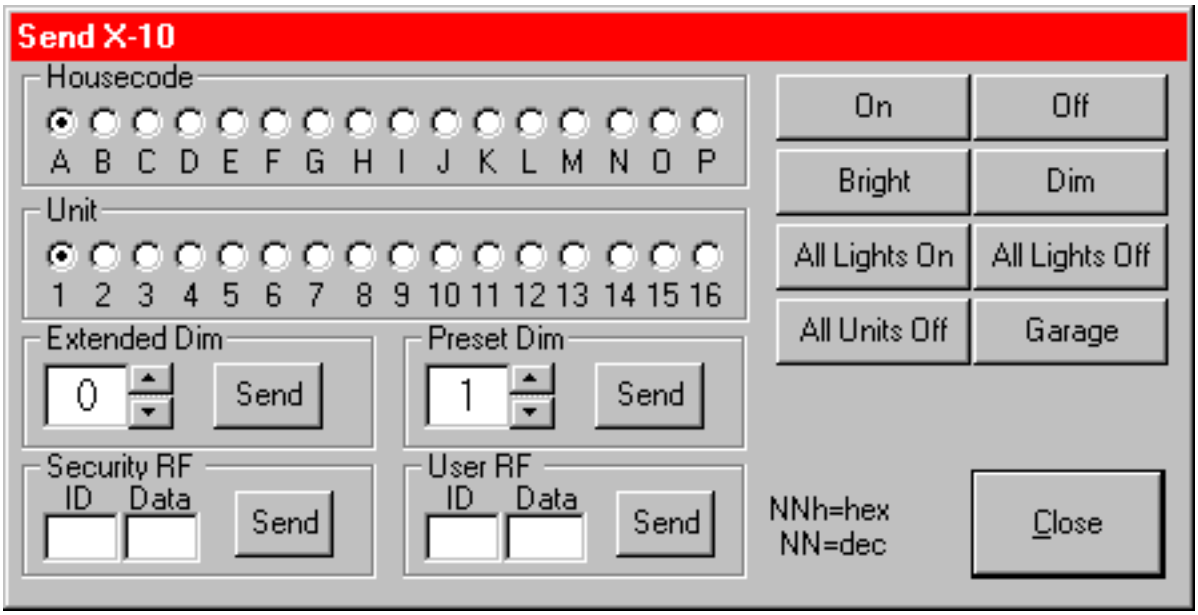
A later release of the firmware will allow downloading Dawn/Dusk tables to the BX24-AHT EEPROM. Set the Longitude, Latitude, and Time Zone for your location. The Longitude and Time Zone values are negative if west of Greenwich, England and positive if east of Greenwich.

RTC Backup



The BX-24 RTC is not battery backed. The BX24-AHT will use the battery backed RTC in the CM11A to sync its internal RTC after any power cycle or watchdog reset.

Direct X-10 Commands



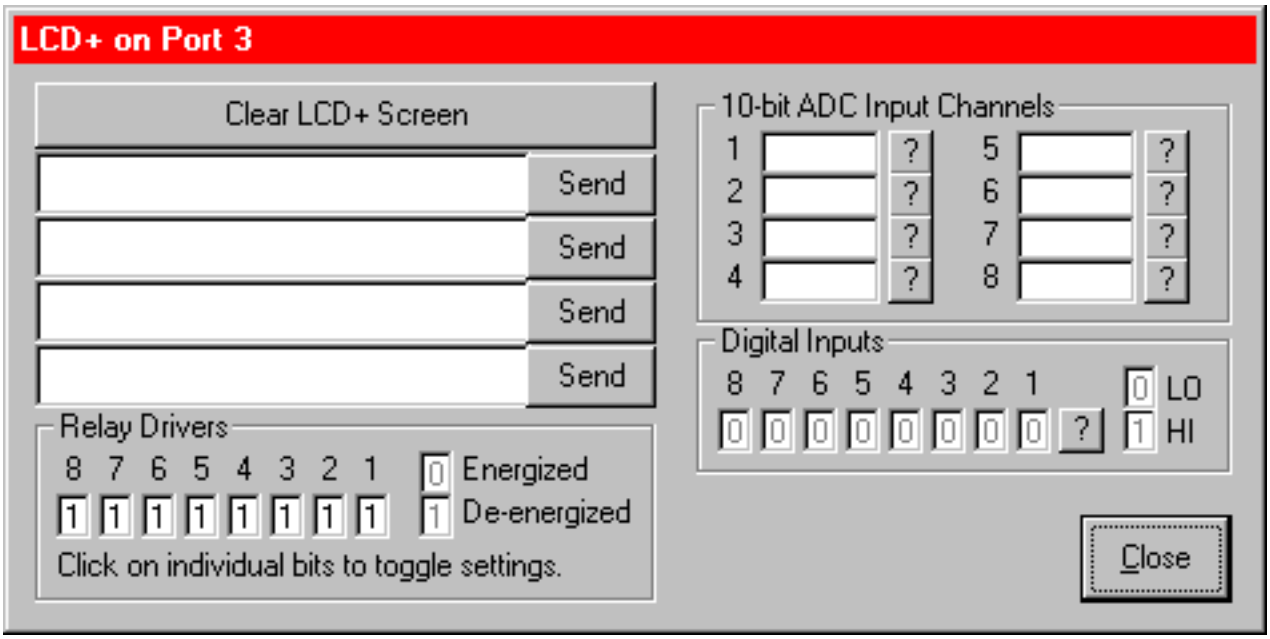
The **Send X-10** panel allows for direct X-10 commands. They are sent by simulating an RF reception. This generates the RF hex codes in the Output Window which can be copied and pasted into **RF Action** definitions.

It also simulates the reception of RF commands for Extended Dims, Preset Dims, and User RF inputs for ADC and Digital values. These are extensions to the X-10 RF protocol that cannot be sent with standard X-10 RF transmitters. They can be sent with an RF-enabled OmniRemote™ Springboard in a Handspring Visor PDA. RFCodes.exe, in the program folder, will generate the codes needed to program the OmniRemote. The User RF commands will also be sent from user constructed RF input transmitters for which we will provide details later.

The RF codes sent by Stanley garage door sensors can also be simulated. SElect the House and Unit codes used and click the Garage button.

Direct LCD+ Commands

The LCD+ is another product from NetMedia that is based on the same Atmel MCU as used for the BX-24. In addition to a 20x4 backlit text LCD, it has 8 10-bit ADC channels, 8 high current relay drivers, and an 8-bit port that can be used for a 4x4 keypad (not supported by the BX24-AHT) or as 8 digital input pins (supported by the BX24-AHT). It is an excellent and reasonably priced way to expand the I/O capacity of the BX24-AHT.



The LCD+ panel can send text to the LCD+, set/clear the LCD+ relay drivers, read the ADC channels and read the digital I/O port.

The BX24-AHT does not have enough EEPROM for string storage except as used with **RF Action** definitions. **Parameter** definitions allow for output to the LCD+ but only of the parameter value, there is no associated text label. You can, however, use the LCD+ panel to send static text labels to the LCD+ and then define **Parameters** to display the values at positions correlated with the static labels. For example, we have **Parameter** definitions that display the date, time, temperature, relative humidity, and barometric pressure on the LCD+ display. We have defined static labels (bold type) such that the final result is...

```
08 DEC 2001     16:36
Temperature:     72.5
 R/Humidity:     86.3
  Barometer:    29.10
```

Note that the LCD+ has no provision for reading the state of the relay drivers. The BX24-AHT keeps track of commands sent to the relays and, when queried, will respond with a value but there is no guarantee that its record is accurate as there may have been power failures, etc. that changed the relay states. If you have no other use for the 8 LCD+ digital input pins, you can connect them to pins 1-8 of the ULN2803A relay driver IC to indicate the state of each relay driver but the only accurate way to indicate actual relay states is to use an extra set of relay contacts to indicate the true state of the relay.

You can set and clear the relays and query their states in **RF Actions** using the ID numbers 037-044 and the command format:

```
~037=0    (sets Relay #1)
~037=1    (clears Relay #1)
~037=?    (returns state of Relay #1)
```

While you can connect more than one LCD+ to the BX24-AHT, only one, the one with the highest port number, can be used with **Parameters**. All you can do with the other is send text to its screen.

Direct ASCII Output

The ASCII panel allows for testing commands to ASCII devices.

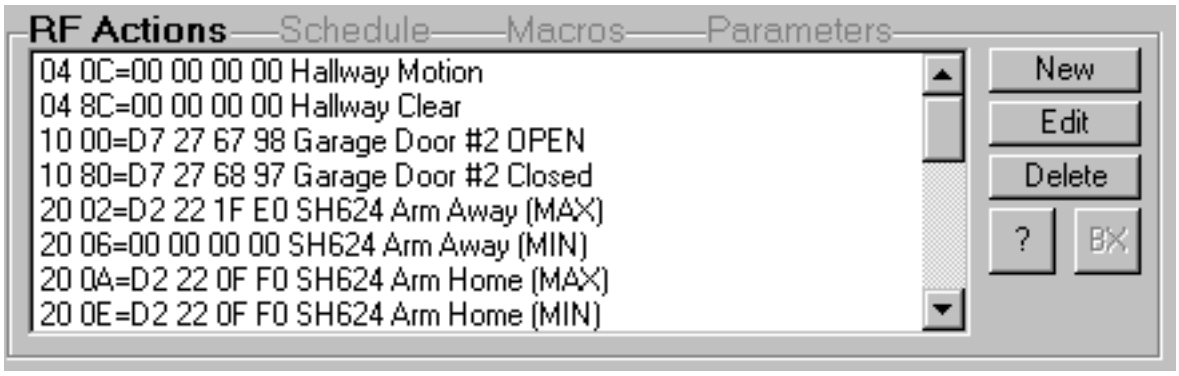
Any device that uses an ASCII protocol at 9600 8N1 can be controlled in this manner using **RF Actions** to trigger the ASCII commands.

Binary commands can be sent using a special format. If an ASCII message contains **~NNN**, the NNN will be translated to the byte value 0-255 denoted by NNN. (NNN must include 3 digits, using leading zeros as needed.) For example:

```
~072~101~108~108~111
```

will send **Hello** to the ASCII device although the primary use is to send non-printing control codes.

Events Window



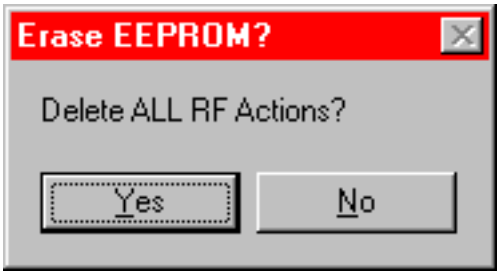
The **Events Window** is for defining and managing the various types of events which, when downloaded to BX-24 EEPROM, control the behavior of the BX24-AHT.

Event types are:

RF Actions	Define how the BX24-AHT responds to received RF
Schedule	Transmit X-10 commands triggered by time of day
Macros	Transmit X-10 commands triggered by received X-10 or, indirectly, by received RF
Parameters	Automatic monitoring of ADC and digital inputs at user defined intervals

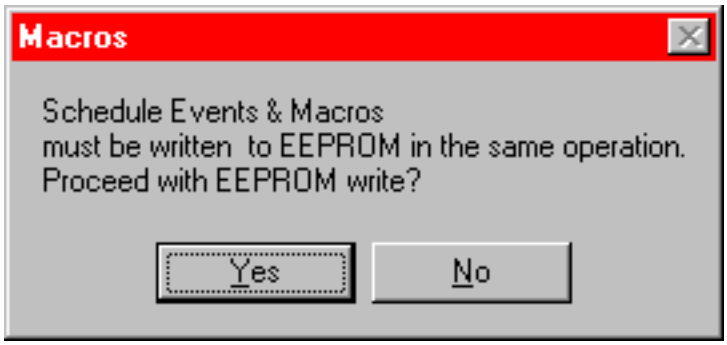
All event types share a common user interface. Click on the type name above the **Events Window** to change the active type. You can add new entries, edit existing entries, delete entries (singly or en masse), view EEPROM contents, and download to EEPROM using the buttons adjacent to the **Events Window**.

To add an entry, click the **New** button. To edit an existing entry, select it with the mouse and click the **Edit** button or *double-click* the entry. With the exception of the last entry, individual entries can be deleted by selecting the entry and clicking the **Delete** button. To delete all the entries (or the last entry) in the active category, *right-click* the **Delete** button and respond to the dialog shown below.



Deleting individual entries requires clicking the **BX** button to download the revised list to EEPROM. Each use of the **BX** button erases and rewrites the entire list. It is best to wait until all revisions are made to avoid excessive erase/write cycles. Deleting all entries by answering **Yes** to the **Erase EEPROM** prompt will automatically reset the BX24-AHT pointers to zero. It is not necessary to click the **BX** button to complete this.

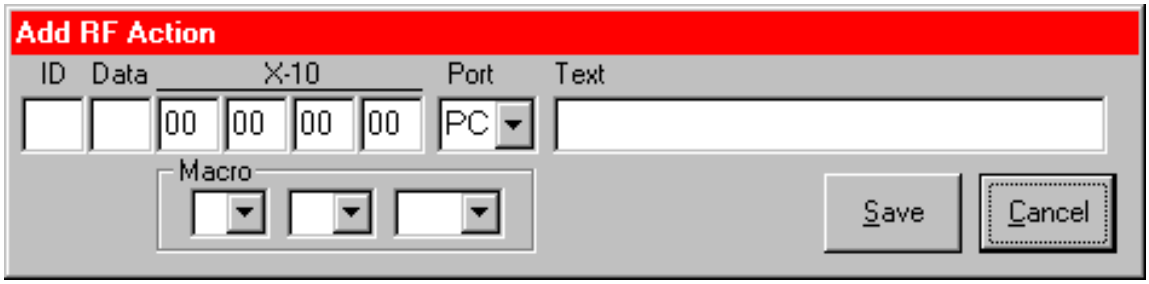
Schedules and **Macros** are intimately related and are stored together in EEPROM. Whenever you attempt to store changes in one category, you will see the prompt shown below. If revisions are also being made to the counterpart category, make those revisions before downloading to EEPROM.



Data entry for the individual event types is detailed in the sections that follow.

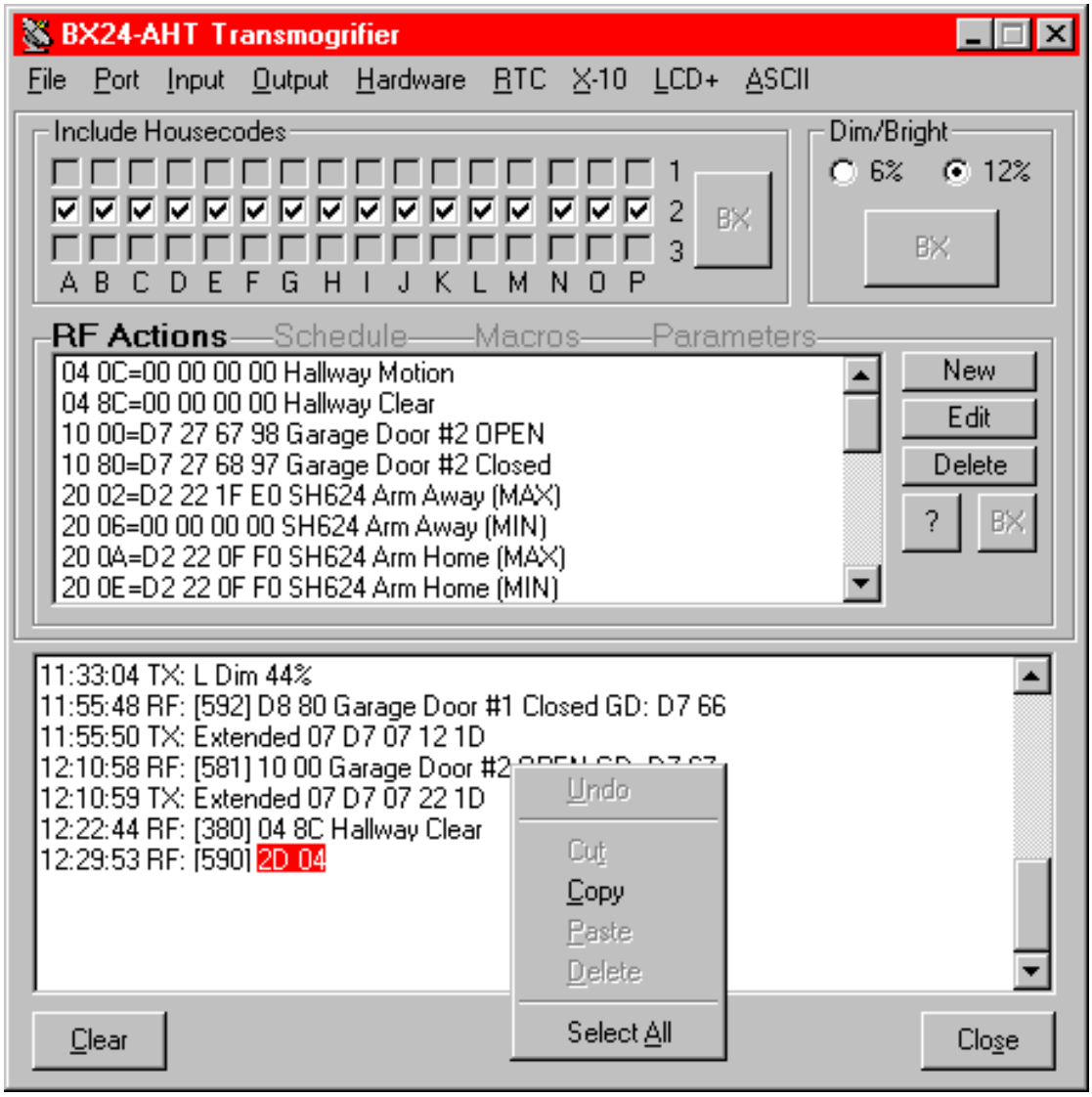
RF Actions

Clicking **New** or **Edit** when the **RF Actions** category is active in the **Events Window** calls the data entry dialog shown below. The only differences between **New** and **Edit** are in the caption and the fact that when editing an existing entry, you will be unable to change the **ID** and **Data** bytes that trigger the action.

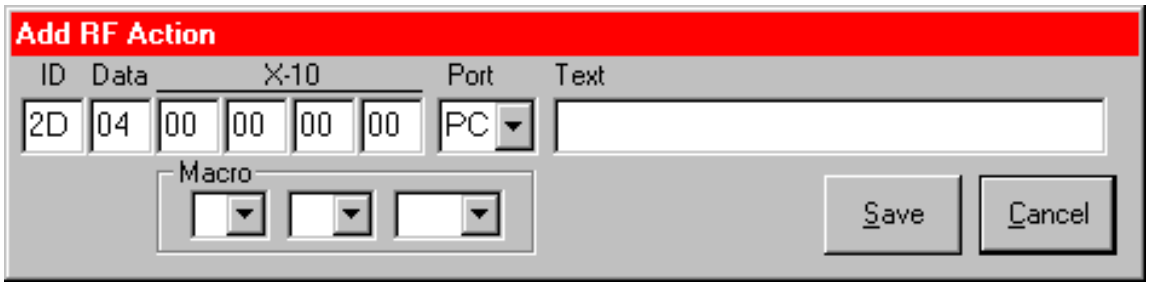


Defining a new **RF Action**, begins with the **ID** and **Data** bytes that will trigger the action. These are the decoded results of a received RF code that are displayed as two hexadecimal bytes in the **Output Window** of the main screen. You can generate these hex bytes by sending the actual RF code with an X-10 RF transmitter or by simulating it using the **X-10 Menu** from the main window. Or you can enter them directly. (*For hex numbers that do not contain a letter, append* ***h*** *to a number to signify hex.*)

Once the decoded RF bytes have been generated, you can highlight them with the mouse. After highlighting them, *right-click* the mouse and copy the codes to the clipboard.

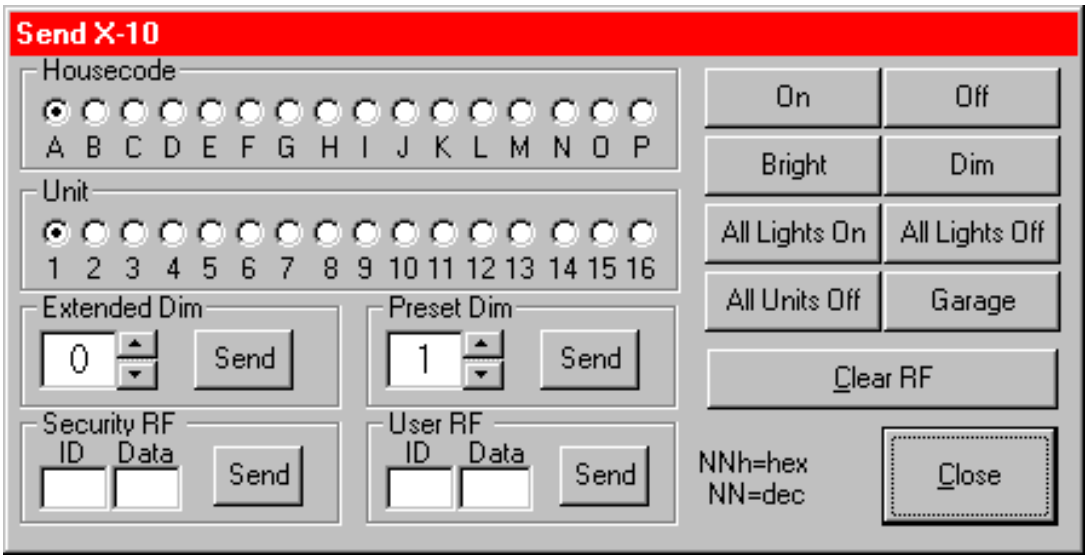


Then click **New** to call the data entry dialog. *Right-click* on either the **ID** or **Data** windows and paste the two bytes from the clipboard.

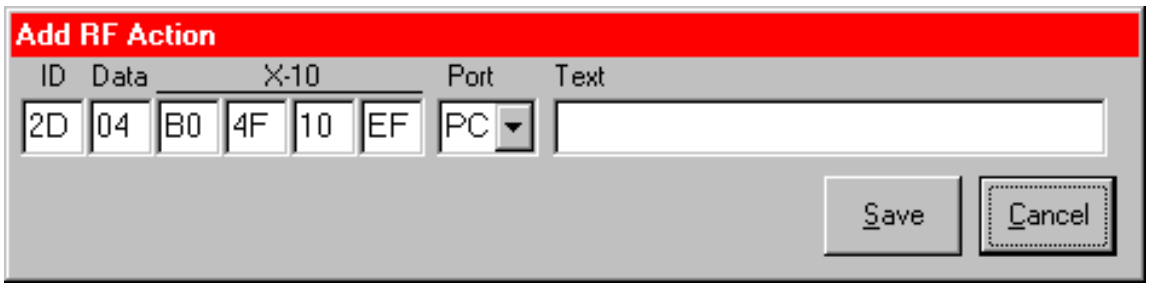


When no **RF Action** has been defined, the default response to received standard X-10 RF is to transmit the X-10 command to the powerline if the housecode is one of the **Included Housecodes**.

As soon as an **RF Action** is defined by entering the **ID** and **Data** bytes, the default action is to do nothing. If you want the received RF command to trigger an X-10 transmission, you must select the X-10 codes to transmit. Clicking on any of the four X-10 windows calls an X-10 dialog from which you can select the X-10 command to transmit to the powerline.



Select the house code, unit code, and function. When you click the function button, the X-10 dialog will close and the four X-10 bytes will be automatically pasted into the data entry dialog.



Instead of transmitting a single X-10 command, you can have the received RF trigger an X-10 **Macro** by selecting the **Macro** trigger. Defining the actual macro commands associated with the trigger is detailed in the **Macro** section.

Note that the Macro section will not be visible unless the four X-10 windows all contain 00.

Click on any of the X-10 windows to call the X-10 dialog and click the **Clear** button to delete the X-10 data. The

macro section will then be visible.

Instead of, or in addition to, any X-10 response to the received RF, you can also enter a text string and select the port to which it will be sent. Selecting the PC port will send the text to the **Output Window**.

Or you can select one of the output ports and send commands to ASCII 9600 8N1 devices attached to those ports. Some examples of text sent to the **Output Window** are:

```
~021=?              (displays ~021=n where n is the value of Parameter 21)
Motion detected     (RF received from an X-10 motion detector)
Kitchen Door OPEN   (RF received from door/window switch)
~019=1              (makes Pin 19 an output and sets it to +5V)
```

Some examples of text sent to other ports are:

```
+X010101            (sends X10 address B2 to an Ocelot)
~016~020Hello       (writes Hello at R2, C1 of LCD+)
~037=0              (sets LCD+ Relay 1 driver low)
```

The BX24-AHT will also receive and decode RF from X-10 security devices. You can define **RF Actions** for them in the same manner as for standard X-10 RF. The only difference being that you cannot simulate those commands from the **X-10 Menu**. Most security devices have a **Test** switch that will send the RF codes when pressed. Then you can copy then from the **Output Window** in the same way as for standard RF.

Schedule

X-10 commands can be sent to the powerline automatically at scheduled times between 0000-2359 with a resolution of one minute. While multiple commands across multiple housecodes can be included in a single schedule entry, it is good practice to keep commands as short as possible to avoid missing RF or other activity. The BX24-AHT can only do one thing at a time and sending multiple X-10 commands can occupy a great deal of time.

To create a new **Schedule** entry, select **Schedule** in the **Event Window** and then click the **New** button to call the Schedule Wizard. To edit an existing entry, *double-click* the entry or highlight it and click the **Edit** button.

The X-10 macro can be up to 16 bytes with 1 byte for each housecode, 1 byte for each address, 1 byte for each function and, for some functions, 1 byte for level. The **Schedule** Wizard guarantees that data is entered in proper order and limits both the total bytes and the options available depending on previous entries. The best way to understand it is to play with it to see what it allows you to enter.

Scheduled events and X-10 Macros are stored together in EEPROM. Whenever you click the BX button to store either, you will see a prompt that reminds you that you need to have both lists ready for storage before proceeding.

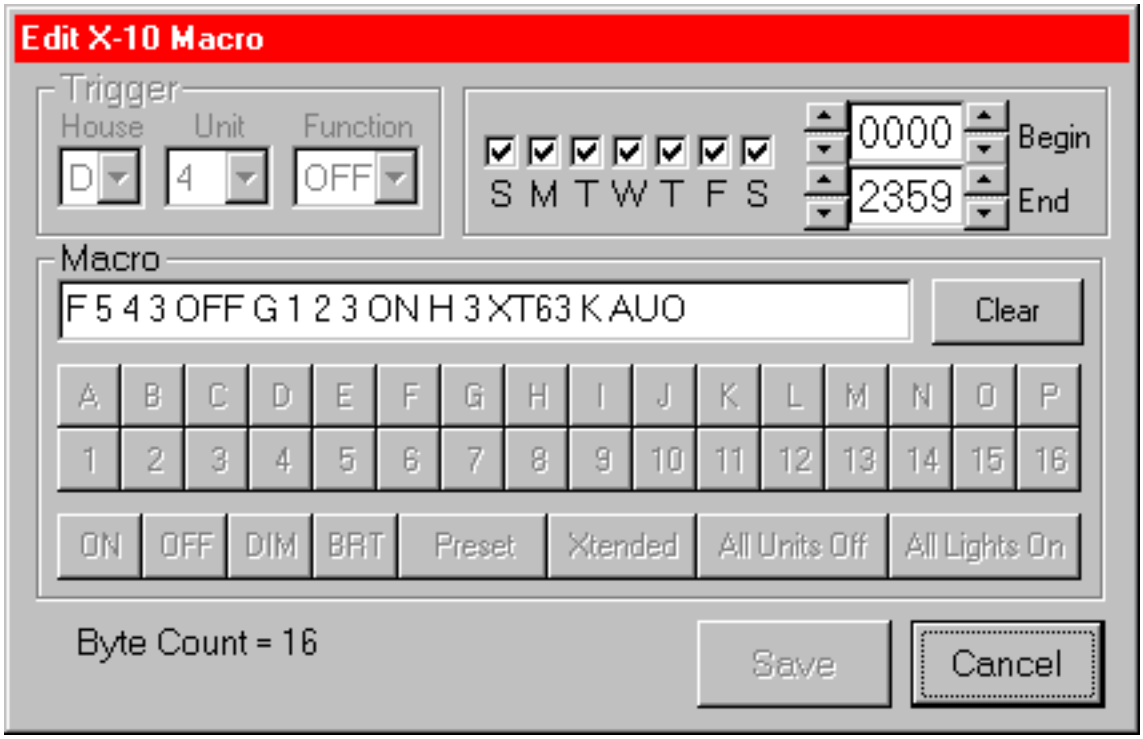
Note: The 0000 events do not execute automatically at all reboots in the current firmware release. This will be enabled once it is clear that the firmware is relatively stable.

Dawn and Dusk lookup tables will also be added in a later firmware release. While you can toggle between Dawn, Dusk, and Time by clicking on the face of the spin control, Dawn and Dusk values have no current meaning to the firmware.

Macros

Macros are similar to **Schedule** events except they are triggered by X-10 On/Off commands *received from the powerline* rather than by time of day. You can also limit a macro to a given time period of the day.

Macros are only triggered when the function immediately follows the trigger address. In the example below, A1 AON will trigger the macro but A1 A2 AON will not.



The Macro Wizard is similar to the Schedule Wizard in the way it guarantees that data is entered in proper order and limits both the total bytes and the options available depending on previous entries. The best way to understand it is to play with it to see what it allows you to enter.

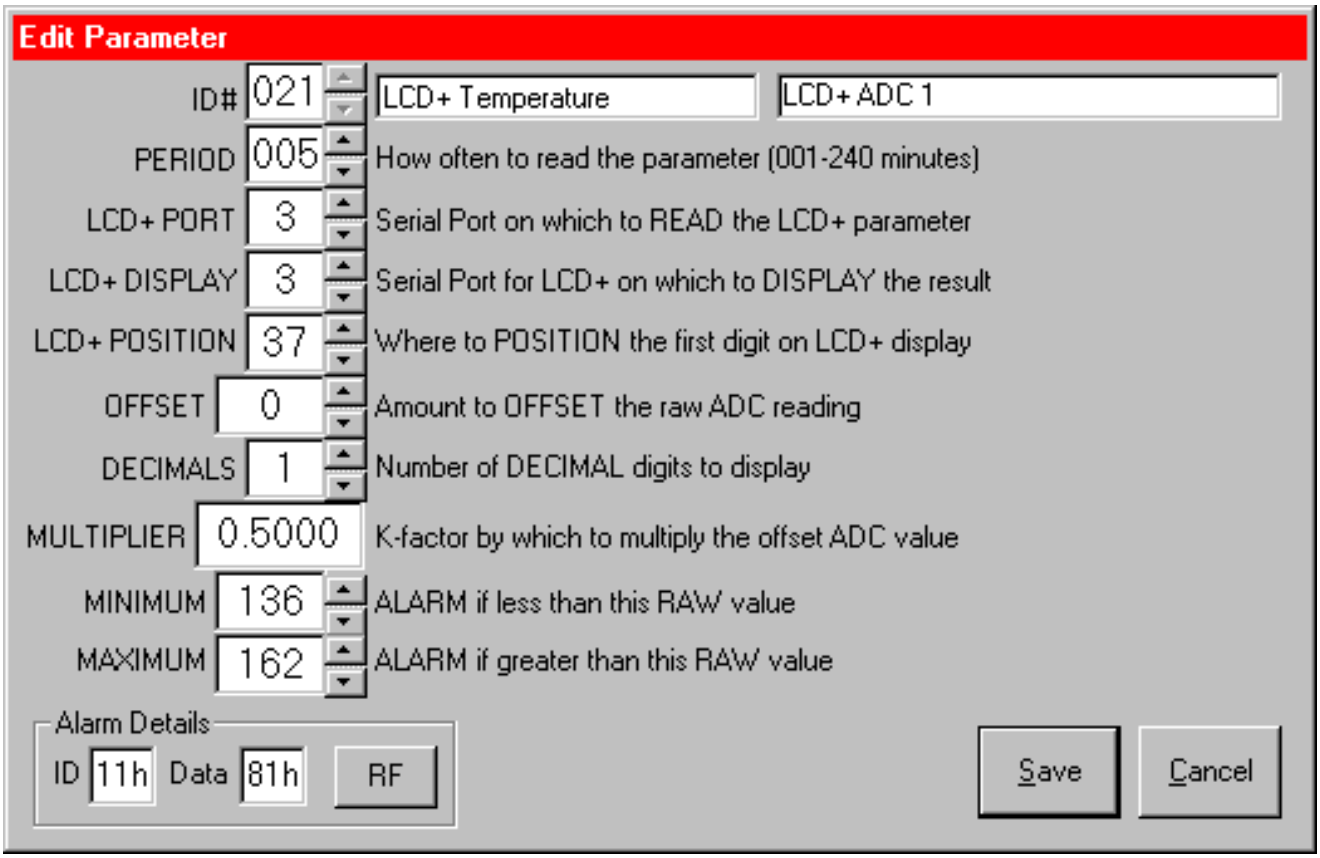
To prevent duplicates, you cannot edit the trigger for an existing **Macro**.

Parameters

Up to 47 **Parameters** can be predefined for the BX24-AHT to read and display the results at predefined periodic intervals ranging from 1-240 minutes. **Parameters** have the preassigned ID numbers listed below.

```
     ID#
       0 RTC Date
       1 RTC Time
       2 Sync RTC to CM11A
     3-4 reserved
    5-19 are for BX24 pins (availability depends on hardware configuration)
      20 reserved
   21-44 are for LCD+ (ADC, relays, digital port)
   45-47 are for Stanley garage doors sensors
   48-50 reserved
  51-100 are for user 12-bit ADC RF nodes
 101-150 are for user 12-bit ADC RS485 nodes
 151-200 are for user digital RF nodes
 201-250 are for user digital RS485 nodes
```

To create a new **Parameter** definition, select **Parameters** in the **Event Window** and then click the **New** button to call the Parameter Wizard. To edit an existing entry, *double-click* the entry or highlight it and click the **Edit** button.



Each **parameter** uses 16 bytes of EEPROM to store the data illustrated in the above example. At each minute rollover, the BX24-AHT scans the parameters list and executes all that are due.

For LCD+ **Parameters** the LCD+ PORT spin control will be enabled. If set to 0, the BX24-AHT will read the

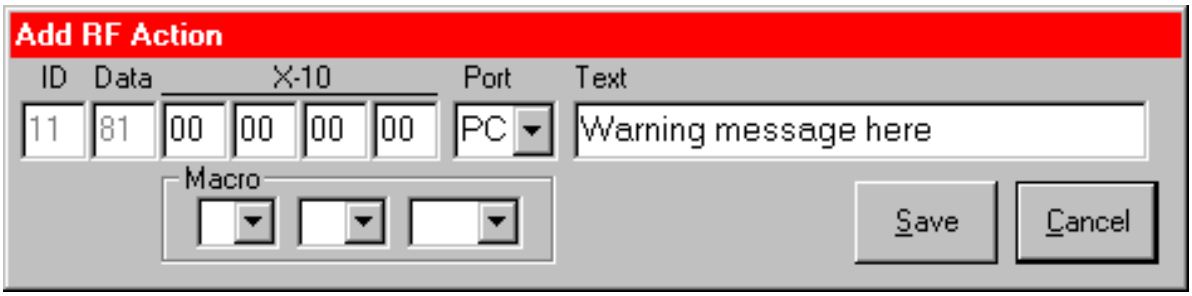
Parameter on the LCD+ attached to the highest output port number. Otherwise, the BX24-AHT will read the **Parameter** from the LCD+ on the port specified.

If LCD+ DISPLAY is set to 0, the result will be sent to the PC port. Otherwise it will be sent to the output port specified.

Although the DISPLAY port is labelled for the LCD+, other devices may be physically connected. If something other than the LCD+ is connected, setting the LCD+ POSITION to 0 will omit leading commands meant to position the LCD+ cursor.

There are few **Parameters** that need monitoring at intervals of less than 5 minutes. We recommend that you should stagger intervals so that the BX24-AHT is not tied up for excessive lengths of time. When it is processing **Parameters**, it cannot be receiving RF, etc.

ADC **Parameters** allow setting high and low limits and defining RF codes that will be *simulated* whenever the Parameter goes out of limits. The *simulated* RF codes use the security RF format. Enter a number between 1-255 (01-FF) for both the ID and Data bytes. (*For hex numbers that do not contain a letter, append **h** to a number to signify hex*.) These can be any arbitrary value as long as they do not conflict with any valid security device codes that may be in use. Once a valid RF code is entered, click the RF button to edit or define an **RF Action** in the same manner as for real RF inputs. Remember to save both the **Parameter** and **RF Action** to EEPROM.



Alarms will be triggered whenever the **Parameter** is out of limits when read. The BX24-AHT will always send **ALARM: ~NNN=n** to the PC and will BEEP the PC.

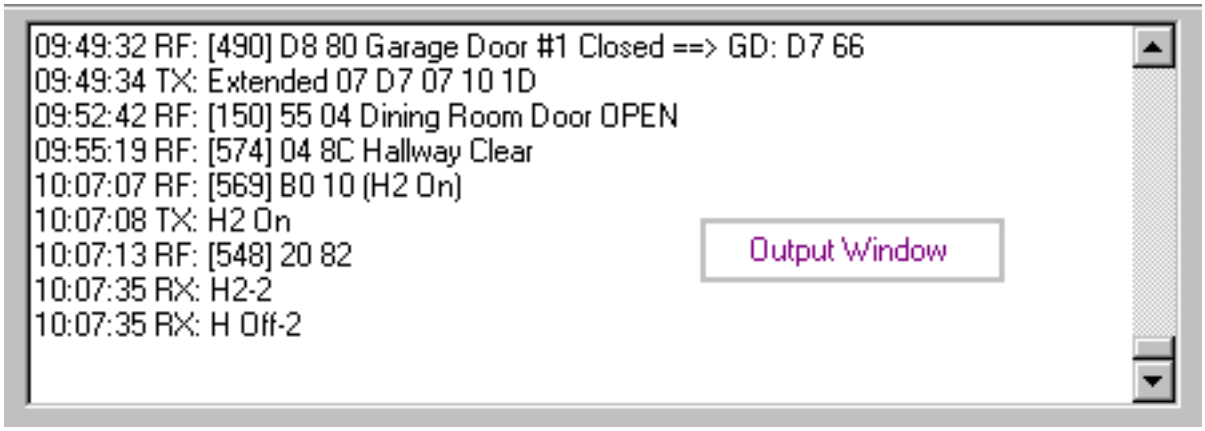
Parameters can also be read on demand with the result sent to the PC or to any output port. This can be done using an **RF Action** or with a serial input (e.g using the command line). The command must begin either with a :n designating the output port or with ~NNN=? when the result is directed to the PC port.

Examples:

```
RF Actions
:3~021=? text goes here
~021=? text goes here
Serial Commands
:3~021=? text goes here|
~021=? text goes here|
```

Output Window

Whenever the BX24-AHT's DB9 RS-232 port is connected to the PC, the **Output Window**, in the lower half of the screen, displays all of the messages sent from the BX24-AHT. Each line follows a similar format, beginning with a timestamp (PC time), which is followed by a 3 character prefix that indicates the source of the data, which is followed by the data.



The character prefixes are:

- RS: Command (received via RS-232)
- RF: X-10 Code (received via RF) with [RSSI*] in brackets
- PS: Preset Dim (received via RS-232 or RF)
- XT: Extended Code (received via RS-232 or RF)
- RA: User RF Analog Data (received via RS-232 or RF)
- RD: User RF Digital Data (received via RS-232 or RF)
- RX: X-10 Code (received from the powerline)
- TX: X-10 Code (sent to the powerline)

  * Received Signal Strength Indicator

For received RF, the decoded ID and Data bytes are displayed as hex bytes immediately after the Recived Signal Strength Indicator. In addition to a digital output, most of the RF receivers that we have tested provide a linear output with amplitude proportional to the received signal strength. The BX24-AHT uses an ADC input to measure this during the long lead-in pulse. It is a reliable indicator of relative signal strength.

Each RX line terminates with a -n where n represents the port to which the CM11A is attached. A future version will support multiple powerline interfaces.

If logging is enabled, this same data is written to BX24-AHT.LOG with a date stamp prepended to each line.

Other Features

RF to IR Converter

The Fire-Stick II from Reynold's Electronics contains a 38KHz oscillator and will output 38KHz modulated IR when a baseband data sequence is input. The RF receivers used with the BX24-AHT all provide a demodulated data output. When a Fire-Stick II is connected to +9V, GND, and the Data line of the RF receiver, it will automatically repeat all RF received by the BX24-AHT as IR.

A later release of RFCodes.exe will be able to convert IR learned by the OmniRemote™ Springboard to 310MHz RF codes. The RF codes can be transmitted by the OmniRemote™ Springboard using the BX24-AHT RF receiver and Fire-Stick II to convert them to IR to control A/V gear.

User RF Inputs

Users can build their own RF transmitter nodes to input temperature and other ADC or digital data to the BX24-AHT. All that is necessary is to use a transmitter that matches the frequency of the BX24-AHT RF receiver and to use the extended RF protocol that we have created for this purpose. There are several manufacturers of 418MHz & 433.92MHz receivers for use outside North America. For North America, Ming makes a SAW controlled 310MHz transmitter which is stocked by Reynold's Electronics. You can also cannibalize the RF transmitter daughterboard from any of X-10's universal remote controls. When Atmel releases their AT86RF401X, which can handle 310MHz, 418MHz, and 433.92MHz, (as well as other frequencies) we will create a universal transmitter design based on it.

The protocol details will be posted to the BX24-AHT web page.

Command Line Interface

The Windows Interface program can receive commands from other Windows applications either through Windows messages to its main window or via the command line. The latter can even receive commands from batch files. See the **Communications Protocol** section for additional details.

Communications Protocol

The PC serial port is 19200 8N1.

The BX24-AHT will recognize the following commands:

```
     XHUUFLL (always 7 bytes)

with X - a prefix that indicates it's X-10
     H - Housecode A-P
    UU - Unit 01-16
     F - Function   U = All Units Off
                    L = All Lights On, Off  (L00 = Off, L01 = On)
                    N = On
                    F = Off
                    D = Dim
                    B = Bright
                    H = Hail Request
                    P = Preset Dim
                    R = Status Request
                    X = Extended Dim
                    M = Macro Trigger
                    - = No Function - Address Only (e.g. XH03-00)
    LL - Level 00-99 for Dim/Bright **
               01-32 for Preset Dim
               00-63 for Extended Dim
               00-01 for Macro Trigger (00 = Off, 01 = On)
               00    for all else
            ** 01-16 for the IP Commander version
```

For the IP Commander version, once the CM11A has sent the command to the powerline, the BX24-AHT will echo the same command. If there is no echo, assume the command was not completed.

Also for the IP Commander version, when commands are received from the powerline, the BX24-AHT will send the same codes with the additions:

```
     XH00S01 for Status On
     XH00S00 for Status Off
     XH00H01 For Hail Acknowledge
```

The BX24-AHT will also respond to:

```
~NNN=0|      echo
~NNN=1|      echo
~NNN=?|      ~NNN=n
The vertical bar terminator is required.
```

where NNN is a 3-digit Parameter ID and n is its current ADC value or Digital state.

For **Parameters**, if the PC port is selected as the display port, the output will be in the ~NNN=n form.

To send text or commands to an LCD+ or other ASCII device connected to the output serial ports, prepend :n where n = 1-3 for the port and terminate the string with a vertical bar.

Examples:

```
:3Your text message here|
:2+V050005|
```

All of the above X-10 commands, when sent to the PC serial port of the BX24-AHT simulate an RF reception so they can be used to trigger RF Actions associated with standard X-10 RF inputs.

Security RF codes and non-Dim extended codes require a different method. For them send the byte value 0x07 followed by the four byte values that represent the four RF bytes.

For example:

```
0x07 0xB0 0x4F 0x10 0xEF
```

will simulate the security RF code sent when I press Panic on my SH624 security remote. The format for security RF codes is explained in the **X-10 RF Protocol** link on my web page at:

http://www.laser.com/dhouston/rf.htm