



# X10 Support User Guide

TRIL Centre's BioMOBIUS™ Research Platform: an Open, Shareable Software and Hardware System

## Disclaimer

The BioMOBIUS X10 components are supplied as is with no warranty implied or otherwise.

## Audience

All

## Purpose

Describes the use of the BioMOBIUS™ X10 related components

## Pre-requisites

BioMOBIUS v2.0 installed.

Version: 1.2

1	Overview .....	3
2	Installation .....	3
3	Design.....	4
4	Configuration and Usage.....	9
4.1	DB_DML.....	10
5	Sample Patch and Database.....	17
6	Troubleshooting.....	<b>Error! Bookmark not defined.</b>

# 1 Overview

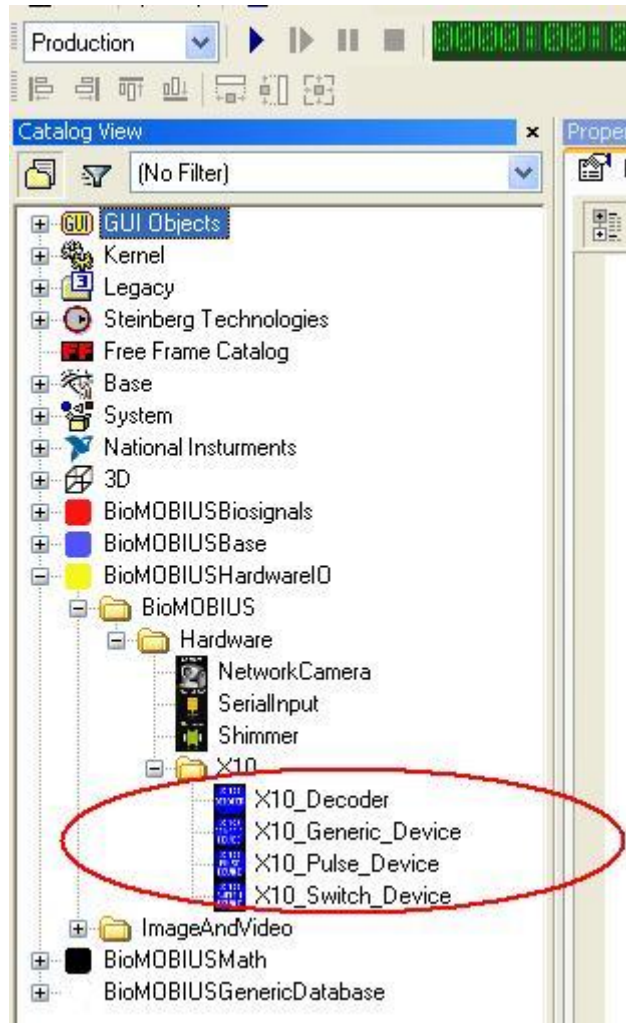
This document describes how BioMOBIUS supports standard off the shelf X10 devices. X10 is widespread - it is used for home security and industrial automation. There are a multitude of X10 devices available and these devices connect over radio frequency (RF) or through the mains. There is also a number of variations in the protocol used to interact with these devices. There are two RF frequencies used – 310MHz (American) and 433MHz (European). This version of the BioMOBIUS X10 support package interfaces with the X10 transeiver over a serial port and is thus independent of the communication mechanism. Those devices which adhere to the standard 32 bit X10 protocol are supported in this version.

The BioMOBIUS X10 support package consists of four basic blocks – an X10 decoder block and three device blocks, sample EyesWeb patches and this document. The blocks enable the user to incorporate simple X10 functionality into an EyesWeb patch without having to develop any code. Most X10 devices are supported by these blocks and those that are not directly supported may be incorporated using the generic device block. X.10 decvices can be added to a patch in just a few moments – the user drags the relevant blocks onto a patch and connects the requisite input and output pins.

The following sections describe the installation, design, configuration and typical use of these blocks. A further section describes a sample patch and a final section details some troubleshooting steps.

## 2 Installation

The X10 components are included as part of the BioMOBIUS v2.0 release and subsequent releases. Figure 1 shows the location of the X10 blocks within the EyesWeb block library.



**Figure 1: X10 blocks in EyesWeb library**

If the X10 blocks are required for BioMOBIUS v1.0, submit a request on [biomobius.trilcentre.org](http://biomobius.trilcentre.org) and a separate catalogue will be made available.

### 3 Design

The blocks are designed to support standard X10 devices as seamlessly as possible. A common decoder block decodes incoming X10 data, extracts the address and decodes the data based on the standard X10 decoding scheme and maps this data to specific X10 devices. The standard X10 addressing scheme based on HouseID and

UnitID is adhered to and data packets are decoded according to the specification:

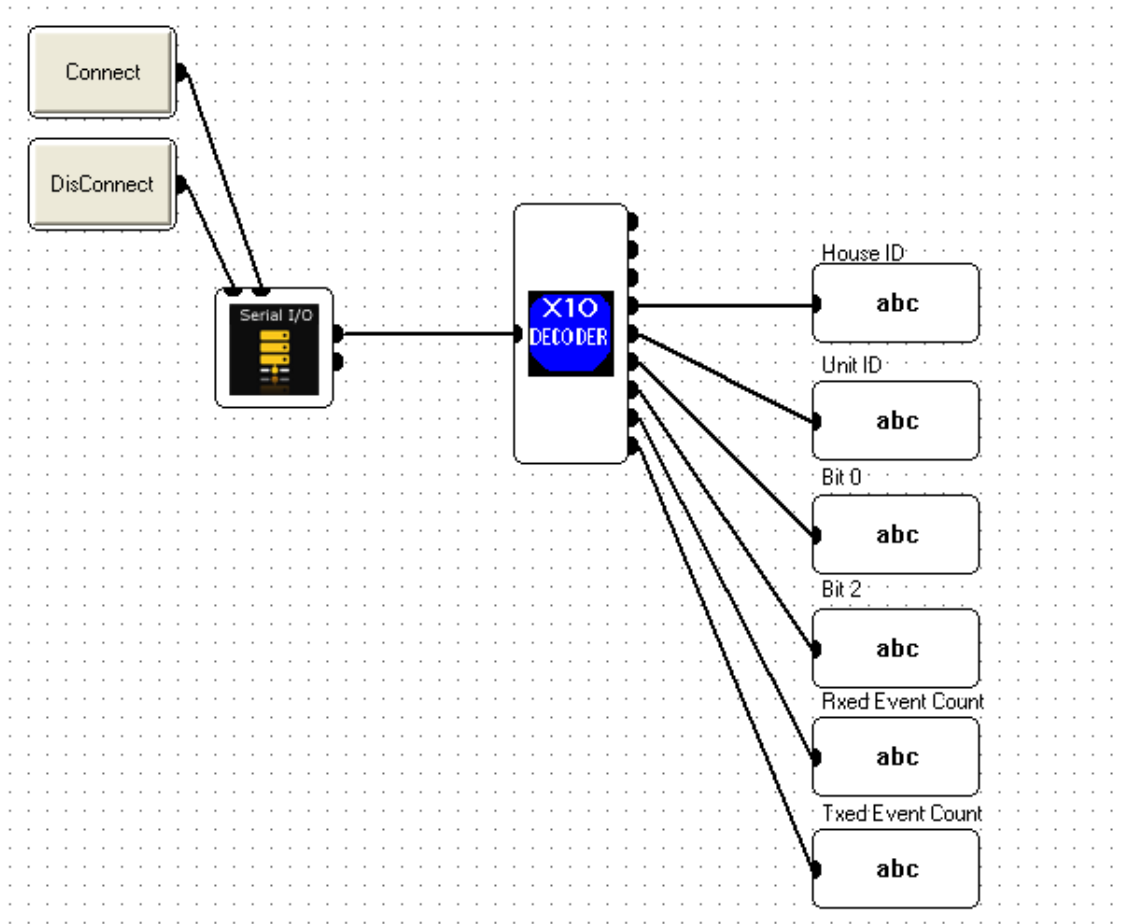
<ftp://ftp.x10.com/pub/manuals/xtdcode.pdf>

The decoder block is independent of whether the X10 devices use the power line or radio frequency as the means of communication, it accepts X10 packets through the standard serial port.

The following usage model describes how one would add X10 devices to an EyesWeb patch.

## 3.1 Usage Model

1. Connect the X10 transeiver hardware to the PC's serial port.
2. Create a patch and add a serial block, an X10 decoder block and string display blocks as shown in figure 1.
3. Configure the serial parameters of the serial block to match the serial characteristics of the X10 transeiver hardware.
4. Run the patch and note the HouseID,UnitID and number of received packets for each X10 device as it is activated.
5. Reconfigure any X10 devices which have duplicate IDs so that all devices report unique IDs..
6. Add X10 device blocks for each deployed X10 sensor choosing the X10 device blocks that best match the X10 sensor. These blocks are described in a later section.
7. Run the patch, fine tune the appropriate block parameters and confirm that the data from each device is decoded correctly.



**Figure 2: X10 device finder patch**

## 3.2 BioMOBIUS X10 Blocks

The objective is to design a series of X10 blocks that offer the following features:

- A mechanism to interpret the X10 addressing scheme and route X10 data to unique down line devices
- Automatically decode the X10 data for the common X10 blocks according to type – e.g. ON/OFF switches, pulsing devices, etc.
- Enable the configuration of operational parameters like duplicate packet handling, in-built timers, etc.
- Add timestamp info to the X10 data.

To meet these objectives the BioMOBIUSX10 catalog consists of the following blocks:

- X10 Decoder – decodes addressing data for all incoming X10 data packets
- X10 Switch Device – decodes X10 data for ON/OFF type devices
- X10 Pulse Device – decodes X10 data for time variant devices
- X10 Generic Device – decodes data for all device types

X10 data is interpreted according to the standard 32bit X10 protocol as specified in this document:

<ftp://ftp.x10.com/pub/manuals/xtdcode.pdf>

A number of features are incorporated in the design as follows:

- Duplicate data suppression – X10 data is generally transmitted in duplicate. Duplicates can be handled at source by the X10 Decoder block or downline in the specific X10 device blocks.
- Common functionality – encapsulate common functionality/features within a block. The switch and pulse blocks are examples of this where support for common data encoding and operational parameters is implemented in common blocks. Consequently, these two blocks should enable the seamless support of the majority of X10 devices.

## 4 Configuration and Usage

The BioMOBIUS X10 blocks are contained in the BioMOBIUSHardwareIO catalog and in the BioMOBIUS.Hardware.X10 library. This section describes how one configures and uses these X10 blocks within the EyesWeb GDE.

## 4.1 X10\_Decoder

### 4.1.1 Block Signature

Label: X10\_Decoder

Catalogs: BioMOBIUSHardwareIO

Libraries: BioMOBIUS.Hardware.X10

### 4.1.2 Block Description

The block accepts X10 data packets from the BioMOBIUS serial driver, decodes the device address and basic data bits, creates a timestamp to record the event, maintains a count of received and transmitted packets, and finally outputs the modified X10 data packet for further decoding by downline x10 device specific blocks.

Normally, X10 devices transmit duplicate data packets in an attempt to overcome inherent data loss. The decoder block provides a simple facility to suppress these duplicate packets – the current incoming data packet is compared with the previously received data packet and if identical then this packet is dropped if configured to do so.

This block is required to carry out the preliminary decoding of the incoming X10 data before passing to specific X10 device blocks.

### 4.1.3 Block Icon



### 4.1.4 Parameters

**SupressDuplicates** – Pass on or suppress duplicate event data. Event data is usually received a number of times. Turning suppression on will cause the decoder to ignore the number of consecutive duplicate packets specified. **DB Type** - Specifies the targeted database engine – e.g. SQLite3, MySQL or MS SQL Server.

**NumberofDuplicates** – The number of duplicate events to suppress. A value of zero specifies that all consecutive duplicate events are suppressed. Typically this is set to a value of 4 - each event data packet is normally received five times.

### 4.1.5 Inputs

**Datastream** – An integer matrix containing the four bytes of the incoming X10 data packet.

### 4.1.6 Outputs

**DecodedData** – an integer matrix containing the decoded data as follows:

- Reverse byte 3 and shift in position of byte 1
- Reverse byte 4 and shift in position of byte 2
- Reverse byte 1 and shift in position of byte 3
- Reverse byte 2 and shift in position of byte 4

**EventTimestamp** – a string representation of the EyesWeb timestamp for the last event

**HouseID** – a single character specifying the X10 HouseID address field.

**UnitID** – integer value specifying the X10 UnitID address field.

**Bit0** – integer value representation of bit 0 of byte 1

**Bit2** – integer value representation of bit 2 of byte 1

**ReceivedEventCount** - Running count of received X10 events regardless of whether duplicate events are suppressed or not.

**TransmittedEventCount** - Running count of X10 events that are passed on, this value is dependant on the NumberOfDuplicates parameter

**EventToggle** - Toggles everytime the block receive an event.

### 4.1.7 Unusual behavior

### 4.1.8 Other Comments

### 4.1.9 Parent

## 4.2 X10\_Switch\_Device

### 4.2.1 Block Signature

Label: X10\_SWITCH\_DEVICE

Catalogs: BioMOBIUSHardwareIO

Libraries: BioMOBIUS.Hardware.X10

### 4.2.2 Block Description

This block processes X10 data for ON/OFF type X10 devices. Typical devices in this category are reed switches and pendants. The block samples the incoming X10 data from the upline X10 decoder block and processes any X10 packets which are addressed for it. The address is determined by the configurable HouseID and UnitID combination.

The incoming X10 data is decoded according to the selected switch type.

### 4.2.3 Block Icon



### 4.2.4 Parameters

**SupressDuplicat**es – Pass on or supress duplicate event data. Turning suppression ON/OFF is dependent on the equivalent setting on the X10 Decoder block.

**NumberofDuplicat**es - The number of duplicate events to supress. A value of zero specifies that all consequetive duplicate events are supressed. Typically this is set to a value of 4 - event data is normally received five times.

**HouseID** – Single character specifying the House ID of this X10 device, valid range is A to P.

**UnitID** - Value specifying the Unit ID of this X10 device - valid range is 1 to 16.

**DeviceType** - Specifies the supported X.10 switch type as follows:

**Standard Switch:** A standard switch setting adjusts the output to follow the value of bit 2 of byte 1. The value is interpreted as 1 = closed and 0 = open.

**Toggle Switch:** adjusts the output on each received packet, for example if the first received packet sets the output to 'ON' then the next packet sets the output to 'OFF' and so on.

**Standard PIR Device:** adjust the output on based on the bit0/bit2 transition, 0/1 is inactive and 0/0 is active.

**ActiveLabel** – specifies the string that is output when the device is deemed to be activated. Typically, this is set to 'OPENED' for a standard switch.

**InactiveLabel** – specifies the string that is output when the device is deemed to be deactivated. Typically, this is set to 'CLOSED' for a standard switch.

#### **4.2.5 Inputs**

**Datastream** – An integer matrix containing the four bytes of the decoded X10 data packet from an X10 Decoder block.

#### **4.2.6 Outputs**

**EventDescription** – **the string** value as specified in the ActiveLabel/InactiveLabel parameters.

**ReceivedEventCount** - Running count of received X10 events addressed for this specific device.

**TransmittedEventCount** - Running count of X10 events that this device passes on.

#### **4.2.7 Unusual behavior**

#### **4.2.8 Other Comments**

#### **4.2.9 Parent**

## 4.3 X10\_Pulse\_Device

### 4.3.1 Block Signature

Label: X10\_PULSE\_DEVICE

Catalogs: BioMOBIUSHardwareIO

Libraries: BioMOBIUS.Hardware.X10

### 4.3.2 Block Description

This block processes X10 data for time variant type X10 devices. A typical device in this category is a passive infra red detector – it responds to initial activation and thereafter resets itself.

The block samples the incoming X10 data from the upline X10 decoder block and processes any X10 packets which are addressed for it. The address is determined by the configurable HouseID and UnitID combination.

The incoming X10 data is decoded according to the selected PIR type.

### 4.3.3 Block Icon



### 4.3.4 Parameters

**SupressDuplicates** – Pass on or supress duplicate event data. Turning suppression ON/OFF is dependent on the equivalent setting on the X10 Decoder block.

**NumberofDuplicates** - The number of duplicate events to supress. A value of zero specifies that all consequetive duplicate events are supressed. Typically this is set to a value of 4 - event data is normally received five times.

**HouseID** – Single character specifying the House ID of this X10 device, valid range is A to P.

**UnitID** - Value specifying the Unit ID of this X10 device - valid range is 1 to 16.

**DeviceType** – Specifies type of pulse device:

- Toggle Device - in this case the output is updated on each event or on the timer firing. Input data values are not sampled.
- STANDARD PIR Device, the output is toggled on either the second event or the expiration of the timer period - whichever is shorter. Activation is determined by `byte1.bit2 = 1`.

**TimerValue** - value specifying the number of seconds to elapse after activation at which point the output is toggled

**FirstEventLabel** – specifies the string that is output when the device is deemed to be activated. Typically, this is set to 'OPENED' for a standard switch.

**SecondEventLabel** – specifies the string that is output when the device is deemed to be deactivated. Typically, this is set to 'CLOSED' for a standard switch.

### 4.3.5 Inputs

**Datastream** – An integer matrix containing the four bytes of the decoded X10 data packet from an X10 Decoder block.

### 4.3.6 Outputs

EventDescription – **the string** value as specified in the ActiveLabel/InactiveLabel parameters.

**ReceivedEventCount** - Running count of received X10 events addressed for this specific device.

**TransmittedEventCount** - Running count of X10 events that this device passes on.

### 4.3.7 Unusual behavior

### 4.3.8 Other Comments

### 4.3.9 Parent

## 4.4 X10\_Generic\_Device

### 4.4.1 Block Signature

Label: X10\_GENERIC\_DEVICE

Catalogs: BioMOBIUSHardwareIO

Libraries: BioMOBIUS.Hardware.X10

### 4.4.2 Block Description

The block samples the incoming X10 data from the upline X10 decoder block and outputs any X10 packets which are addressed for it. The address is determined by the configurable HouseID and UnitID combination.

The X10 incoming decoded data is passed unmodified to the output pin – it is the responsibility of downline blocks to process the data.

### 4.4.3 Block Icon



### 4.4.4 Parameters

**SupressDuplicates** – Pass on or supress duplicate event data. Turning supression ON/OFF is dependent on the equivalent setting on the X10 Decoder block.

**NumberofDuplicates** - The number of duplicate events to supress. A value of zero specifies that all consequetive duplicate events are supressed. Typically this is set to a value of 4 - event data is normally received five times.

**HouseID** – Single character specifying the House ID of this X10 device, valid range is A to P.

**UnitID** - Value specifying the Unit ID of this X10 device - valid range is 1 to 16.

**DeviceType** – Specifies type of generic device. Currently this defaults to the Generic X10 Device.

### 4.4.5 Inputs

**Datastream** – An integer matrix containing the four bytes of the decoded X10 data packet from an X10 Decoder block.

### 4.4.6 Outputs

**DecodedData** – an integer matrix containing the unmodified decoded X10 data

**ReceivedEventCount** - Running count of received X10 events addressed for this specific device.

**TransmittedEventCount** - Running count of X10 events that this device passes on.

### 4.4.7 Unusual behavior

### 4.4.8 Other Comments

### 4.4.9 Parent

## 5 Sample Patch

The X10 package contains a sample patch:

- X10BlockTest.eywx

This patch demonstrates the reception of X10 encoded data from the standard BioMOBIUS serial block, decoding of this data and the connection of downline switch and pulse blocks.

This patch may be used initially to decode and identify the addresses of X10 devices and then incorporate support for these devices by simply adding additional switch/pulse blocks and linking these to the output pin of the decoder block.

The X10\_Generic\_Block may be used to support devices which do not conform to the standard switch/pulse type, e.g. temperature sensors.

Please contact BioMOBIUS support if you wish to add support for additional device type in future BioMOBIUS releases.

