

# Applied Digital Ocelot Plugin for HomeSeer

## *Change Log*

### **Changes for 2.1.0.0**

- Compatibility changes for HomeSeer 2

### **Changes for 1.0.33**

- Fixed IR addresses (locations) above 383.

### **Changes for 1.0.32**

- now supports more than 32768 for variables
- auto detects 512 or 1024 IR locations

### **Changes for 1.0.28**

- removed log entry filling up ocelot.log file

### **Changes for 1.0.27**

- added support for the Leopard touch screen. See below for details

### **Changes for 1.0.26**

- calling cpuxa.SetVar now updates variable in HomeSeer properly

### **Changes for 1.0.25**

- fixed cpuxa.LearnIRStart was not accessible from HomeSeer

### **Changes for 1.0.24**

- on fast variable changes, HS could become out of sync with real variable values

### **Changes for 1.0.23**

- fixed a problem with 1.0.22 (released with 1.6 of HomeSeer) where variable last change times keep changing
- fixed a problem with 1.0.22 where events would not trigger on variable changes
- changed variable handling to decrease CPU usage

### **Changes for 1.0.20**

- removed some debug statements from the log

### **Changes for 1.0.19**

- SECU16-IR zone 17 is the infrared emitter on the Ocelot unit
- fixed possible page fault with bad data returned from Ocelot unit

### **Changes for 1.0.18**

- when deleting devices, if only units are re-created, the relay outputs are not controllable
- parameter 7 in Ocelot is set to 0 (no need to rescan for new modules)
- when plugin is used as IO only, comport setting was not being read from INI file
- SECU16-IR zone 0 was not operational from HomeSeer events

### **Changes for 1.0.15**

- setting ocelot variable values over 256 now set properly

### **Changes for 1.0.14**

- added warning to user if Ocelot is not added as I/O device when vars and I/O points are used

### **Changes for 1.0.13**

- added support to store config information in settings.ini file in HS config directory (feature added in HS beta 38)
- removed some debug statements from the log

### **Changes for 1.0.12**

- previous change to I/O input detection was reporting incorrect input point
- added support for SECU16-IR from GUI

### **Changes for 1.0.11**

- New method for detecting I/O point changes was implemented this should make I/O point detection mode reliable
- SECUI and RLY8 modules are now fully supported

### **Changes for 1.0.10**

- timeouts and CRC COMM errors are only logged if two of the same errors occur sequentially
- added communications debug option to the setup form

### **Changes for 1.0.7**

- increased polling timeout from 2 to 4 seconds

## Changes for 1.0.6

- now polls for IO incase input or output changed without notification
- properly supports up to 31 units

## Changes for 1.0.3

- now reads variables properly with CMAX executive 2.98 (cmax 1.7)

## Changes for 1.0.3

- on startup the status of digital inputs/outputs are updated
- now supports up to 21 external units

## Changes for 1.0.2

- more fixes to remove checksum errors

## Changes for 1.0.1

- fixed checksum errors during IR sending
- fixed IR sending slow
- added check so multiple copies of the exe cannot run

## *Requirements*

- HomeSeer Version 1.5.0.0 or higher must be installed.

## *Installation*

1. To install the Applied Digital Ocelot plug-in for HomeSeer version 2.x, select the “Updater” button on the web interface, or select “Updates” from the “Help” menu. Follow the on-screen instructions for the Updater.
2. Select the Applied Digital Ocelot plug-in on the package selection screen, then follow the on-screen instructions to finish the update process, and restart HomeSeer.
3. After restarting HomeSeer, click the “Setup” tab and then the “Interfaces” tab on the web site. Click “Disabled” to enable the plugin, and then click “Save” at the bottom of the screen.
4. Enable I/O, IR, or X10 based upon what technologies you are using.

## *Leopard Support*

Starting with version 1.0.26 Leopard support for the touch screen has been added.

The following script commands allow you to write data to the Leopard screen.

```
WriteScreen X,Y,txt,backlight,font,clear,beep
```

Where:

Parameter	Description
X	X coordinate to start writing
Y	Y coordinate to start writing
txt	the text to write, up to 40 characters
backlight	TRUE to turn on the backlighting, FALSE = turn it off
font	1 = large font 0 = small font
clear	TRUE = clear the screen
beep	TRUE = trigger the beeper

To write text on the screen use the following format in a script:

```
hs.plugin("Applied Digital Ocelot").WriteScreen 0,0,"hello on the leopard screen",true,0,false,false
```

To clear the screen:

```
hs.plugin("Applied Digital Ocelot").WriteScreen 0,0,"",true,0,true,false
```

Sample script to fill the screen with text:

```
Sub Main()
' clear screen
hs.plugin("Applied Digital Ocelot").WriteScreen 0,0,"",true,0,true,false
y = 16
for i=1 to 14
hs.plugin("Applied Digital Ocelot").WriteScreen 0,cint(y),"hello on the leopard screen",true,0,false,false
y=y+16
next
end sub
```

The Leopard can also be put into a mode so that screen touches are reported.

You can use this 2 ways. It can be set to return the button that was pressed, or the screen coordinates. You can set the mode with the following commands:

```
' set it to return the screen coordinates
hs.plugin("Applied Digital Ocelot").EnableTouch 2

' set it to return the button press
hs.plugin("Applied Digital Ocelot").EnableTouch 1

' turn the response off
hs.plugin("Applied Digital Ocelot").EnableTouch 0
```

The plugin now has a new button on the config screen, "Create Leopard Device." Click this button to create a new HS device that will hold the information returned when a user touches the screen.

For example, if the user presses the lower right hand corner (location 59), the value 59 will be stored in the device "Leopard Screen". You can now create an event in HS to trigger when HS detects a value change on this device. Just use the Value Change trigger. I also modified HS with 2 new value change options (Set To) and (Set To Any). The "Set To" options allow the event to trigger even if the same value is set into a device. This is useful in this case where you may want to detect the user pressing the same button, and the value 59 is repeatedly set into the device.

To check the screen coordinates in a script, use the Value Change trigger, and set the type to "Set To Any". The event will trigger with any touch to the screen. In the script, use the `hs.DeviceValueByName("Ocelot Leopard Screen")` to get the coordinate number.

To change which screen is displayed on the Leopard, set variable 63 to the screen number like:

```
hs.SetDeviceValueByName "Ocelot Variable 63",2 ' show screen 2
```

## ***Support***

For Support:

Help Desk: [www.homeseer.com](http://www.homeseer.com) (select ***HelpDesk*** from the ***Support*** menu)

Support is also available on our message board at:

<http://board.homeseer.com>

Look for the forum dedicated to the Applied Digital Ocelot plugin.