



# Intel Unite® Plugin Reference Design for Camera Sensor

---



## **Legal Disclaimers & Copyrights**

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com](http://intel.com).

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel, the Intel logo, Intel Unite, Core, and vPro are trademarks of Intel Corporation in the United States and/or other countries.

\*Other names and brands may be claimed as the property of others

© 2015 Intel Corporation. All rights reserved.



# Table of Contents

---

1	Camera Sensor Reference Design .....	4
1.1	Description .....	4
1.2	Requirements .....	4
1.3	Project Structure and Setup .....	4
1.4	Build and Installation .....	5
2	Configuration .....	5

# 1 Camera Sensor Reference Design

## 1.1 Description

This plugin reference design is an example to demonstrate how to control the monitor using the Intel Unite solution and a USB camera to detect room occupancy. When the camera detects occupants in the room, or people are connected to the Intel Unite solution, the monitor will be enabled. After a configurable amount of time of no occupancy detected, the monitor will go to sleep.

This plugin uses the OpenCvSharp library to detect motion from the camera.

## 1.2 Requirements

- Visual Studio 2015 or greater.
- Intel Unite solution v2 through v3.x.
  - CFCPlugin.dll – Information on referencing this abstract library can be found in the Intel Unite SDK guide.
- OpenCvSharp: This component is a 3<sup>rd</sup> party library that is used to detect motion from a USB camera.
  - This project is setup to download this dependency from NuGet. If you are unable to get this dependency from NuGET you will need to download, build, and update the references in this project.

## 1.3 Project Structure and Setup

This project has 4 main files which are required to build the project.

### *Config.cs*

This file contains the Config class, which is used to serialize and de-serialize the settings contained in the MotionDetectPluginSettings.xml. This class also contains helper functions to assist with settings and business logic.

### *MonitorControl.cs*

This file contains the MonitorControl static class, which is used to set the power state of the monitor. This imports user32.dll and exposes two methods: *Wake()* and *Sleep()*.

### *MotionDetectPlugin.cs*

This file contains the main plugin which the Intel Unite app will load, *MotionDetectPlugin*.

When the Intel Unite app loads on the hub, it will execute *Load()*, which will initialize the plugin. As part of the initialization, this plugin will load a thread which will run the *MotionDetectPlugin::DetectMotion* class. This class uses the OpenCV library to detect motion using the USB Camera. This class also handles calling the monitor *Wake()* and *Sleep()* methods.

### *MotionDetectPluginSettings.xml*

This XML file contains the configuration options for this implementation of the plugin. A full description of options can be found in the Configuration section of this guide.

## 1.4 Build and Installation

You should be able to open the solution, build, and deploy this reference design without making any modifications to the source. To do this, ensure that you have Visual Studio 2015 or greater installed and that you have access to get NuGet packages.

- 1) Open **MotionDetectPlugin.sln** using Visual Studio
- 2) Note the **Configuration Manager** build type (Debug or Release)
- 3) Select **Build** then **Build Solution**

Visual Studio should automatically fetch OpenCvSharp from NuGet, and the build should complete successfully.

To install, copy the build assets from the bin\\(Debug | Release)\\ to the plugin folder in the hub installation folder (Default: C:\\Program Files (x86)\\Intel\\Intel Unite\\Hub\\Plugins).

## 2 Configuration

You can fine tune the behavior of the plugin by updating MotionDetectPluginSettings.xml, located in the Plugin folder on the hub.

The configuration XML is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Settings>
  <CAMERA_INDEX>0</CAMERA_INDEX>
  <MONITOR_SLEEP_TIME>1</MONITOR_SLEEP_TIME>
  <IGNORE_CONNECTED_USERS>true</IGNORE_CONNECTED_USERS>
  <SHOW_UI>true</SHOW_UI>
</Settings>
```

The CAMERA\_INDEX field indicates which camera is used for detecting motion. "0" indicates default camera will be used to detect motion.

MONITOR\_SLEEP\_TIME field indicates the amount of idle time in minutes after which the monitor is put to standby mode.

IGNORE\_CONNECTED\_USERS flag indicates if the monitor should be put into standby mode irrespective of the users (clients) connected to hub or not.

SHOW\_UI flag indicates whether an Icon will be displayed on hub to indicate the state of motion.

The configuration file can be enhanced to add additional parameters that were used to identify if there is motion or no motion.