

# LiuXiaoDong's Weight of Insomnia

## USER MANUAL AND DOCUMENTATION

by Fito Segrera

In this document you will find instructions on how to install and configure the camera, server, and computer vision application of the robotic-painting machines developed for LiuXiaoDong's "Weight of Insomnia".

### 1. Bill of Materials

#### 1.1. Camera:

- Raspberry Pi 3 x1
- Camera Module V2 x1
- MicroSD Card ( $\geq 16$ GB class 10) x1
- Surveillance Camera Case for Outdoor use x1(big enough to put the RPi and camera module inside)
- Power Adapter(microUSB port, 5VDC 2A) x1
- RJ45 (Ethernet) Cable x1
- Internet access to the server

#### 1.2. Server:

- Online virtual server with public IP address capable of running nodeJS on terminal. An example of a service with these characteristics is LINODE (<https://www.linode.com/>).

**Important Note:** Please be aware that if the piece will be shown in China, the service that you choose to run the server should NOT be blocked by the Chinese fire-wall.

#### 1.3. Computer Vision PC:

- Intel NUC (good specs for running openCV in openframeworks)

**Important Note:** In the past we have used UBUNTU GNOME as the OS and openframeworks to program the APP. If in the future you decide to change this you will have to take care of all appropriate changes by yourself.

#### 1.4. Camera Vision Real-time Monitoring:

- Raspberry Pi 3 x1
- Power supply for Raspberry Pi 3 x1
- 16 GB or bigger micro SD card class 10(fastest) x1
- $>22''$  Display with 3 graphic interfaces (HDMI)

**Important Note:** Each robotic painting machine has 2 Intel NUCs and 1 Raspberry Pi that need to be visualized on the display. The display should have 3 different interfaces, hopefully all HDMI, if not possible, then make sure you have the right adapters from the different possible image signals. For example, if the screen has 1 HDMI and 2 VGA, then you can use the HDMI for the Raspberry Pi and get DV to VGA adapters for both intel NUCs.

## 2. Setup

### 2.1. Camera Setup

The camera is basically a Raspberry Pi 3 with a camera module and some custom scripting in NodeJS.

#### 2.1.1. Camera Hardware

The raspberry Pi and camera module should be mounted inside a weather-proof surveillance camera case.

#### 2.1.2. Camera Software

The software for the camera is all contained and ready in a raspbian image file which can be burn to a microSD card and booted in a raspberry pi 3. The image file comes ready with all necessary scripts and auto-run applications, basically it is ready to plug and play. Just a few minor changes have to be done in order for the online server to recognize each camera as a different ID.

To reconfigure a camera after burning the image file into the microSD card and inserting it into the Raspberry Pi 3 please follow these steps:

**a)** Plug the raspberry pi to its power supply and connect a HDMI screen, keyboard and mouse. You could also access the raspberry pi via SSH from another computer on the same network, for the raspberry pi needs to be plugged to the router via Ethernet cable, more instructions here: <http://www.instructables.com/id/Use-ssh-to-talk-with-your-Raspberry-Pi/>

**b)** Once you are on the raspberry pi, open a terminal window and run the following command:

```
crontab -u pi -e
```

It will ask you which editor you wish to edit with. Select nano. You will then see a line that looks like this:

```
@reboot /usr/bin/sudo -u pi -H /usr/local/bin/forever start  
/home/pi/remotePaintingMachine/camera/app.js cam1
```

You only have to replace *cam1* for the ID of that raspberry pi. The possible IDs are:

```
cam1  
cam2  
cam3
```

Each of this corresponds to 1 of the 3 different cameras.

**c)** Go to the “remotePaintingMachine/camera” folder located in the home folder of the Raspberry Pi and open the app.js file with a text editor. Make sure you first modify the IP address to POST images to your own server. Change the following line (replacing <http://chronusartcenter.org> with your server's address):

```
var url = "http://chronusartcenter.org:3333/stream?id=" + id;
```

## TO MANUALLY SETUP AND INSTALL THE CAMERA SOFTWARE ON A FRESH INSTALL OF RASPBIAN IN THE RASPBERRY PI 3, PLEASE FOLLOW THESE INSTRUCTIONS:

**a)** From the provided source code files, copy the folder called "remotePaintingMachine" to the home directory in the Raspberry Pi 3.

**b)** Make sure that NodeJS and NPM are installed on your raspberry pi:

```
node -version  
npm -version
```

If these are not installed, please follow these instructions: <https://blog.wia.io/installing-node-js-on-a-raspberry-pi-3>

**c)** Open a terminal on the Raspberry Pi and go to the camera folder:

```
cd remotePaintingMachine/camera/
```

Then run the following command:

```
sudo npm install
```

**d)** After all npm dependencies have been properly installed, run the following command:

```
crontab -u pi -e
```

It will ask you which editor you wish to edit with. Select nano. You will then see a line that looks like this:

```
@reboot /usr/bin/sudo -u pi -H /usr/local/bin/forever start  
/home/pi/remotePaintingMachine/camera/app.js cam1
```

You only have to replace *cam1* for the ID of that raspberry pi. The possible IDs are:

```
cam1  
cam2  
cam3
```

Each of this corresponds to 1 of the 3 different cameras.

**c)** Finally open the app.js file with a text editor. Make sure you first modify the IP address to POST images to your own server. Change the following line (replacing <http://chronusartcenter.org> with your server's address):

```
var url = "http://chronusartcenter.org:3333/stream?id=" + id;
```

## 2.2. Server Setup

### 2.2.1 Server Software

After you have registered or setup your online server, please follow these simple instructions:

- a)** Make sure nodeJS and NPM are installed on your online server.
- b)** Copy the folder “remotePaintingMachine” to the home folder on your server.
- c)** Navigate into the “remotePaintingMachine/server/” directory and run the following command:

```
sudo npm install
```

- d)** Make sure you install forevejs on your server:

```
sudo npm install -g forever
```

- e)** Run the server, within the “remotePaintingMachine/server/” directory, and keep it running while the server is up and online:

```
forever start sever.js
```

## **2.3. Computer Vision PC Setup**

These instructions are for an Intel NUC PC running UBUNTU GNOME. Similar steps should work with Windows and/or OSX.

### **2.3.1. Software Setup**

- a)** Make sure you download and install openframeworks:

<http://openframeworks.cc/download/>

<http://openframeworks.cc/setup/linux-install/>

- b)** Download all required openframeworks’ addons to the “openFrameworks/addons” directory:

```
ofxLibwebsockets  
ofxOpenCv  
ofxSimpleTimer
```

You can find all addons at: <http://openframeworks.cc/addons>

- c)** Create a new openframeworks project with the project generator, call it as you wish and make sure you include all the required addons mentioned above. Finally please copy ALL (.h and .cpp) source files from the “lxd\_v1\_4\_no\_gui/src/” to your project’s “src/” folder. Also you need to copy the config.xml file from “lxd\_v1\_4\_no\_gui/bin/data/config.xml” to your project’s “bin/data/” directory.

- d)** Make sure you configure the app in the config.xml file which looks something like this:

```
<CONFIG>  
  <interface>wlp2s0</interface>  
  <server>http://chronusartcenter.org:3333</server>  
  <imageTypeIndex>5</imageTypeIndex>  
  <camera>cam1</camera>  
  <width>600</width>
```

```
<height>500</height>
<contrast>0.6</contrast>
<threshold>30</threshold>
<backgroundUpdate>1800000</backgroundUpdate>
</CONFIG>
```

We can understand this configuration like this:

**interface:** what you are using to connect to the internet.

note: in linux and mac use the terminal command "ifconfig" to see the interface.

**imageTypeIndex:** Integer between 1 and 5

- 1 -> original image
- 2 -> Gray image
- 3 -> High Contrast image
- 4 -> Blobs image
- 5 -> Contours View image

**camera:** cam1, cam2 or cam3

cam1: Beijing  
cam2: Shanghai  
cam3: lxd hometown

**contrast:** value between 0.0 and 1.0

**threshold:** value between 0 and 100

**backgroundUpdate:** CV reference frame

long type 1800000 is default in milliseconds for an update every 30 mins