

# CS CAPSTONE INSTALLATION GUIDE

APRIL 16, 2019

## A-FRAME LIVE STREAM PORTAL

PREPARED FOR

OREGON STATE UNIVERSITY

BEHNAM SAEEDI

PREPARED BY

GROUP 48

NEXUSPHERE

MEGHAN MOWERY

LOUIS DUVOISIN

SARAH Pelayo

### Abstract

This is our installation guide for our A-Frame Live Stream Portal project. This document lists the hardware and software that will be needed. It also describes how to install the software on to the hardware. The installation guide does not go into detail about how to use the system because it is out of this documents scope.

## CONTENTS

<b>1</b>	<b>Materials</b>	<b>2</b>
<b>2</b>	<b>Raspberry Pi setup</b>	<b>2</b>
2.1	The Pi Itself . . . . .	2
2.1.1	Installing picam . . . . .	3
2.1.2	Setting up picam server . . . . .	3
2.2	Video streaming . . . . .	4
2.3	Port forwarding . . . . .	4
<b>3</b>	<b>Server</b>	<b>4</b>
3.1	Upload files . . . . .	4
3.2	Create a database . . . . .	5
3.3	Additional code . . . . .	5
3.4	Creating the device . . . . .	6
<b>4</b>	<b>Conclusion</b>	<b>6</b>

## 1 MATERIALS

Materials that we used for this project are as follows:

- 1) Raspberry pi 3 b+
  - 2) Raspberry pi camera module
  - 3) 1 monitor
  - 4) 1 mouse
  - 5) 1 microphone
  - 6) 1 keyboard
  - 7) HDMI cable
  - 8) Power cable
  - 9) 360 lens
- Note: for this project we used the Kogeto Dot 360 lens for the iPhone.

## 2 RASPBERRY PI SETUP

### 2.1 The Pi Itself

- 1) Install whatever version of raspbian you would like on your raspberry pi (for this project we used raspbian stretch), more information can be found here: <https://www.raspberrypi.org/downloads/raspbian/>
  - 2) Plug it into the peripherals (monitor, keyboard, mouse, etc...).
- Note: you may need to plug in the peripherals before turning on the raspberry pi.
- 3) Follow the setup that the raspberry pi asks for, set the time zone and make sure you are connected to the internet, then update the pi.
  - 4) Find the IP address of the pi by either running 'ifconfig' on the pi's terminal or hovering over the internet connection symbol on the top right of the screen.
  - 5) Run 'sudo apt-get install npm'.
  - 6) Set the raspberry pi to receive video
    - a) enter: 'sudo raspi-config' into the command line
    - b) select 'Interfacing options'
    - c) select 'camera'
    - d) select 'yes'
    - e) select 'OK'
    - f) select 'Finish'
    - g) select 'reboot'
  - 7) The pi will then restart with the camera option enabled.

### 2.1.1 Installing picam

- 1) Run: `sudo apt-get update`
- 2) Run: `sudo apt-get install libharfbuzz0b libfontconfig1`
- 3) From the 'Script' directory in the GitHub repository, move the 'make\_dirsh.sh' script to your pi's main page.
- 4) Note: or you can enter it manually by entering:

```
cat > make_dirs.sh <<'EOF'
#!/bin/bash
DEST_DIR=~ / picam
SHM_DIR=/run/shm

mkdir -p $SHM_DIR/rec
mkdir -p $SHM_DIR/hooks
mkdir -p $SHM_DIR/state
mkdir -p $DEST_DIR/archive

ln -sf $DEST_DIR/archive $SHM_DIR/rec/archive
ln -sf $SHM_DIR/rec $DEST_DIR/rec
ln -sf $SHM_DIR/hooks $DEST_DIR/hooks
ln -sf $SHM_DIR/state $DEST_DIR/state
EOF
```

- 5) Run: `chmod +x make_dirs.sh` to add permissions to run the script.
- 6) Run the script: `./make_dirs.sh`
  - Note: if you want to adjust the sound run: 'alsamixer'
- 7) Run: `wget https://github.com/iizukanao/picam/releases/download/v1.4.7/picam-1.4.7-binary.tar.xz`
- 8) Enter: `tar xvf picam-1.4.7-binary.tar.xz`
- 9) Enter: `cp picam-1.4.7-binary/picam ~/picam/`
- 10) Enter: `'cd ~/picam'` to move to the picam directory.
- 11) To test that the camera is running correctly, run: `./picam -alsadev hw:1,0` and you should see these lines:
 

```
configuring devices
capturing started
```
- 12) Return to the home directory before moving to the next section.
  - Note: If you experience any errors, check that all lines of code were typed correctly without any spelling or grammatical errors.

### 2.1.2 Setting up picam server

- 1) Enter: `sudo npm install coffee-script -g`
- 2) Enter: `git clone https://github.com/iizukanao/node-rtsp-rtmp-server.git`
- 3) Enter: `cd node-rtsp-rtmp-server`

- 4) Enter: `npm install -d`
- 5) Enter: `cd ~/node-rtsp-rtmp-server`
- 6) In `http.cookie`, add `'header += "Access-Control-Allow-Origin: *\n" '` under line 108.
  - Note: Make sure that port forwarding is properly set up between your pi and the router before you begin streaming, check section 2.3 for details.

## 2.2 Video streaming

- 1) In one terminal enter:
  - `cd ~/node-rtsp-rtmp-server`
  - `./start_server.sh`
- 2) In a second terminal enter:
  - `cd ~/picam`
  - `mkdir ~/node-rtsp-rtmp-server/public/picam/`
  - `mkdir ~/node-rtsp-rtmp-server/public/picam/stream`
  - `./picam -alsadev hw:1,0 -rtspout -w 800 -h 480 -v 500000 -f 20 -hlsdir ~/node-rtsp-rtmp-server/public/picam/stream`
  - Note: If you have errors when starting the stream, make sure that the camera module is properly plugged into the raspberry pi and the pi is configured to receive video (see section 6 in 2.1 for more details).

## 2.3 Port forwarding

The node server on the pi expects to get requests on port 80, which is troublesome if you have multiple Pis streaming. You'll need to open the router settings on the network where you plan to be streaming from. Typically the IP for this will be 192.168.1.1 but this may change router to router.

- 1) On the Raspberry Pi type `'ifconfig'` in a terminal window and find the IP address starting with 192 (this is the IP of the pi on the local network).
- 2) In the router settings forward whatever external port you want (typically above 1023) to port 80 of the Raspberry Pi's IP. Look up how to do this for your router as it varies make to make.
- 3) Go to a computer and type `"what is my ip"` in to google and get your ipv4 address, write it and the external port you chose in step 2 down, we'll need them later to add a device.

## 3 SERVER

### 3.1 Upload files

- 1) First, download all of the code from our GitHub repository to your own computer: <https://github.com/BeNsAeI/ASP-IOT>
- 2) Put the `Code` directory in your `public_html` directory on the OSU servers.
  - Note: the files under the `'Code'` directory will be used for the website and the `'Script'` directory will be used to execute the stream commands.

- 3) Put the following bash script from the 'Script' directory (named permissions.sh) in the Code directory in your public\_html if you don't want to change every permission manually. Make sure you change the permissions of the script itself in order to run it. You can also enter it manually:

```
#!/bin/bash
chmod 751 *.php
chmod 771 database.php
chmod 775 css
chmod 775 css/*
chmod 775 images
chmod 775 images/*
chmod 775 js
chmod 775 js/*
chmod 771 errors
chmod 771 errors/*
```

You may see some errors about certain files not existing (particularly errors/\*) these will go away as you use the website.

### 3.2 Create a database

Note: if you already have your own database you can simply import the sample-db.sql file and skip this step.

- 1) Set up your ONID database, log in to [onid.oregonstate.edu](https://onid.oregonstate.edu) with your ONID login information.
- 2) Click on the "Web Database" link on the left side of the page, create the database if needed.
- 3) Write down or take a picture of the database settings table, we'll need it in a future step. (You don't need to write it down if you just keep the tab open).
- 4) Open PHPMyAdmin here: <https://secure.oregonstate.edu/onidddb> and login with the database username and database password (NOT ONID).
- 5) Click on the current database link on the left side of the page (should be your database username).
- 6) Click the import tab and upload the 'sample-db.sql' file found in the main page of the GitHub repository.
- 7) Click on either the devices or map tables on the left just to check if they exist and have data.

### 3.3 Additional code

- 1) Create a file called config.php in your Code directory.
- 2) Enter the following code in to config.php (don't copy paste, some characters like quotes and dashes will break):

```
<?php
global $databasepass;
$databasepass = 'XXX';
global $databasehost;
$databasehost = 'XXX';
global $databasename;
$databasename = 'XXX';
```

```
global $databaseuser;
$databaseuser = 'XXX';
?>
```

- You'll need to enter your own database password, host, name, and user from the previous section in to the respective fields.
  - Note: follow section 3.2 if you don't have a database setup
- 3) After the file is written and saved, run the bash script (e.g. permissions.sh) again to ensure all the files have proper permissions.
  - 4) The webpage should now be viewable on `http://web.engr.oregonstate.edu/~ONIDUSERNAME/Code/index.php`
    - If you downloaded the GitHub repository directly to the server the url might be `/ASP-IOT/Code/index.php` instead.
  - 5) Log in using either the "admin" or "viewer" tokens.

### 3.4 Creating the device

- 1) Once everything is installed and streaming, you can add a device to the website.
- 2) Go to the website, and log in as an admin using the "admin" token.
- 3) Click the add device button, pick a name (alphanumeric), code (alphanumeric), and map coordinates (1 indexed) for the device.
- 4) Enter the IP and Port of the raspberry pi. Check the port forwarding section for more information.
- 5) Click on the device you just made and watch the stream.
- 6) Note: adding a device to the website is gone over in detail in the 'Operation' manual.

## 4 CONCLUSION

This concludes the instillation for the A-frame live stream portal project.