



runlinc Online Project A1: IoT Camera Streaming (runlinc Online Version)

Contents

Introduction.....	1
Part A: Getting to know runlinc	4
Part B1: Preparation (Android)	5
part C1: Program and run the Project (Android)	6
Part B2: Preparation (iPhone)	8
Part C2: Program and run the Project (iPhone)	13

This project uses runlinc online which can be found here:

<http://www.runlinc.com/online/control.html>

Make sure to right click the link and click 'open in new tab'.

Check that your board IP starts with `http://` not `https://`, as shown to the right.

Board IP:	<code>http://runlinc.com</code>
-----------	---------------------------------

This project does not require a runlinc Wi-Fi Inventors' Kit.

Introduction

Aim

This project teaches you how to include vision in your runlinc projects by live-streaming a video from your phone's camera or any other IP camera that can connect to WIFI and stream it to runlinc.

Background

Streaming is the idea or method of transmitting and/or receiving data continuously over a network. Streaming is used in many applications with the most notable being multimedia. This is due to multimedia requiring a large amount of data for a computer to play the media. But then, you might think, isn't downloading and streaming the same thing?

They are similar indeed, but one significant

difference is that downloading requires you to have the file first downloaded into your device's memory or storage before the file can be used which in many cases will take up a significant amount of storage on your device. Whereas streaming does not need to store the files in their entirety, it breaks the files into small chunks which are downloaded, viewed and then deleted once they aren't needed anymore. This also speeds up the viewing as the start of the media can be viewed while later portions are still being downloaded.

This can have implications for devices with small memory. Video files can often have megabytes of data for a few seconds of recording, devices such as runlinc on STEMSEL with only a small amount of flash memory, streaming is necessary. But the downside of streaming is that since the data is not stored, the data must be streamed again if it is required a second time.

Cameras can be used remotely by streaming data to a host. In this project, we'll utilise this application to turn a camera into a remote camera. Camera streaming has security and safety applications such as checking your home when a sensor has sensed someone is in the house when you are not there; and with digital processing, it has applications in AI and robotics like image recognition.

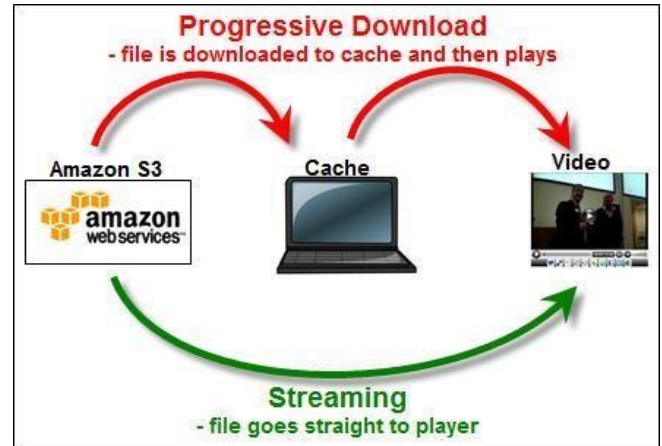


Figure 1 A visualization between download vs streaming.

Plan

In this project, we can set up security cameras to protect your property using the camera of your phone, even give a new life to your old phones and use them as the camera and then stream the data from the camera to runlinc to show the video on the camera live.

runlinc Background

runlinc is a web page inside a Wi-Fi chip. The programming is done in the browser and sent to the chip over Wi-Fi. The runlinc web page inside the Wi-Fi chip will command the microchips to do sensing, control, data logging Internet of Things (IoT). It can predict and command.

Part A: Getting to know runlinc

Before we start programming, we need to open runlinc online. You will find this at <http://runlinc.com/online/control.html>. Make sure you right-click on the link and select 'open in new tab' so you still have this document open. You will be greeted with the following webpage.

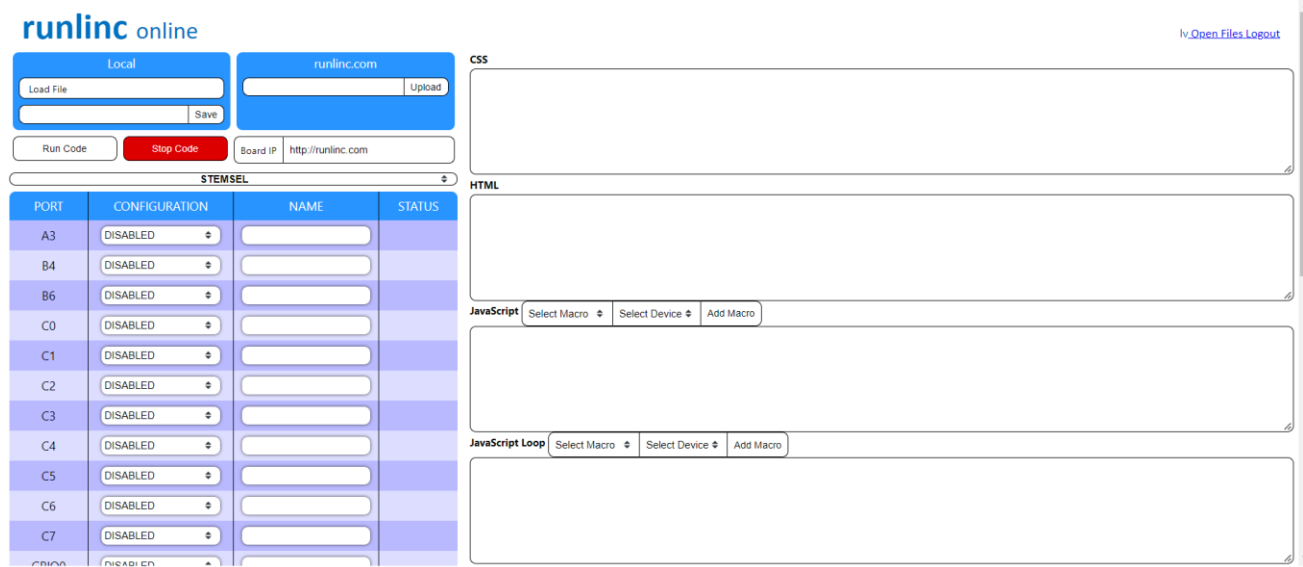


Figure 2 runlinc online webpage

At the top left, there are options to load or save a file, buttons to run and stop the code, and a field labelled 'Board IP'. Make sure the board IP is 'http://runlinc.com' as shown. Some browsers put https, which won't work. If your browser does this, try clearing your site, or browser data. The main section on the left is the board interface which is used for setting up pins on a STEMSEL board. This project doesn't require the board, so we will not be using this section.

The main sections that we will be using are the code boxes on the right. CSS is used for customizing the look of the web page, HTML is for the things you will see on the web page, and JavaScript is for programming the control, sensing, and data logging actions.

The webpage you create will be shown in the preview box at the bottom of the page once you click 'Run Code' up the top.

Part B1: Preparation (Android)

The app you will need to download from the app store is called “AirDroid”. It is free to download.

Once AirDroid is installed and opened, tap on the “AirDroid Web” button. This will display 2 options. Under option 2, you will find the IP Address you will need to use. (As shown below in figure 3).

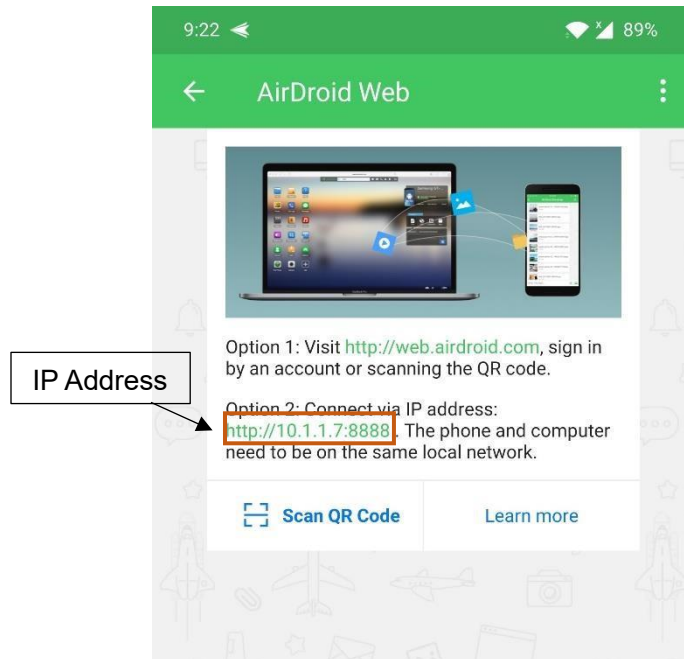


Figure 3 Expected view of IP Webcam (Android).

Note: Make sure both the browser device and the phone are on the same local network. Make sure there is only one camera connection at a time.

In your web browser type in the IP address shown below in the url box of a new tab to check that it works. You should see a desktop screen as shown in figure below.



Figure 4 AirDroid desktop view

part C1: Program and run the Project (Android)

These are the only two lines of code that we will need for this project. One line for a heading, and one line to display our camera feed. This needs to go in the HTML box as shown in figure 4. Replace the asterisks with the IP address we got in the previous step.

```
<h2 style="text-align: center">Robot Camera View</h2>

<iframe style="position: absolute; height: 100%; width:100%; border: none"
src="http://****:8888/"></iframe>
```

The screenshot shows the runlinc online interface. At the top left, there are two file managers: 'Local' and 'runlinc.com'. Below them are buttons for 'Run Code' and 'Stop Code', and a 'Board IP' field containing 'http://runlinc.com'. A table labeled 'STEMSEL' has columns for PORT, CONFIGURATION, NAME, and STATUS. The HTML editor on the right contains the following code:

```
<h2 style="text-align: center">Robot Camera View</h2>

<iframe style="position: absolute; height: 100%; width:100%; border: none"
src="http://10.1.1.7: 8888/"></iframe>
```

Figure 5 HTML in runlinc online

The first line will add a heading to your page, it will display 'Robot Camera View' in bold. The second will create a container which displays the camera feed. The 'src' attribute tells the iframe where the content is coming from, so make sure that URL is replaced with your IP address from figure 2 (using the IP shown in figure 3 the URL would be: <http://10.1.1.7:8888/>).

Click 'run code' at the top left of your runlinc page. You should get something like we have in figure 6 below. Click on the camera icon in the bottom left corner to start streaming the camera (you might be prompted with some popups on your phone to allow the app to use the camera).

If you encounter any jerking of the image close the app on your phone and then re-load it. This will refresh the connection and should remove the image jerking around or not syncing.

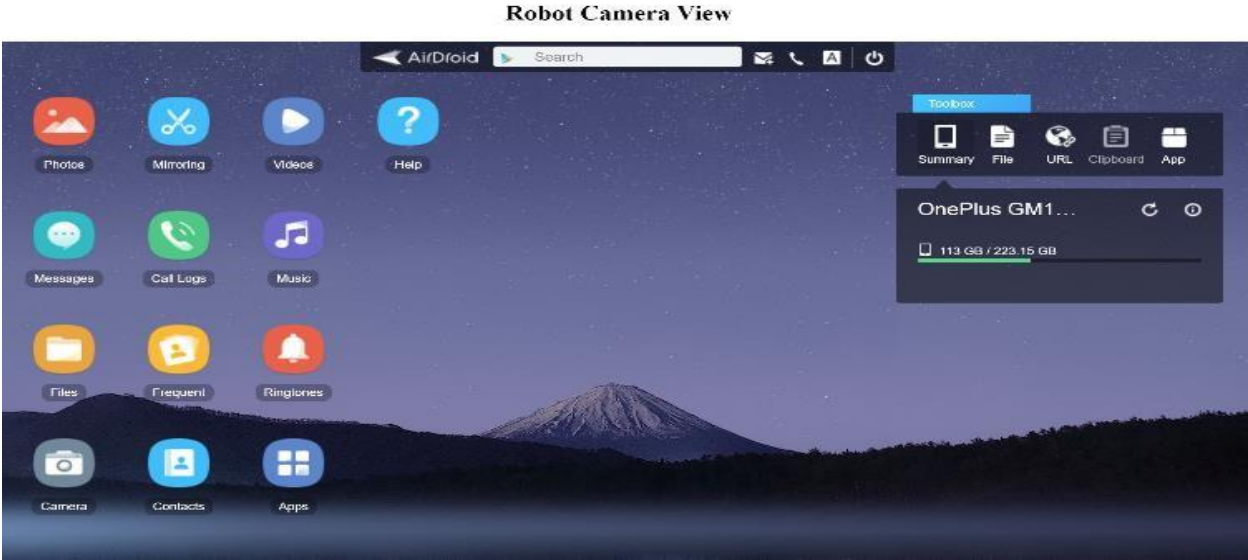



Figure 6 Expected webpage

Part B2: Preparation (iPhone)

Note: Make sure both the browser device and the phone are on the same local network.
Make sure there is only one camera connection at a time.

1. The App that you need to download is “IP Camera Lite”  , free from the iOS App store. There are other apps that you can use if you wish, but this was the one tested.

2. Once downloaded, open the App, and allow the application to access your camera. Allow to find and connect to devices on your local network.

Before activating your camera, press the three-lines button on the top right of the screen. And then tap on Settings.

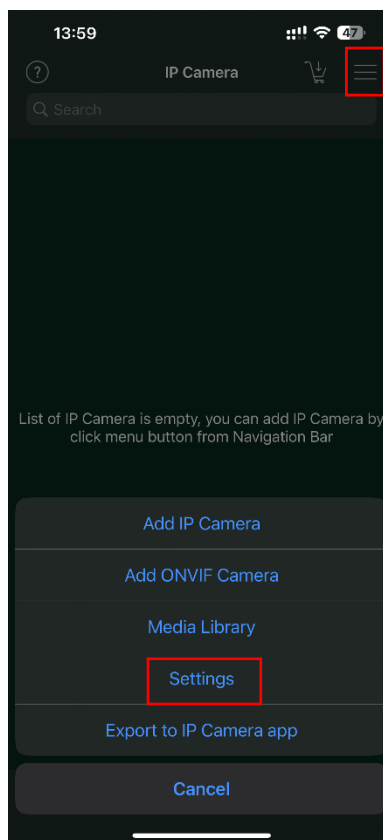


Figure 7 screenshot of interface of IP Camera Lite

3. After opening Settings, scroll down and find the interface shown in the figure below. You can edit the User and Password as you want.

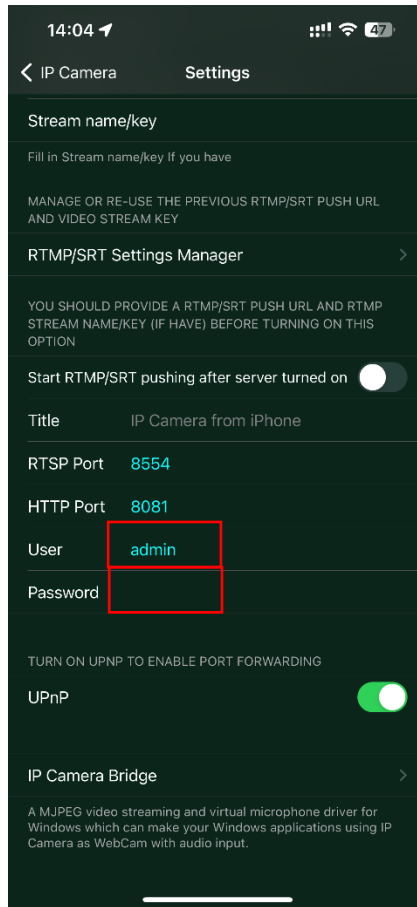


Figure 8 screenshot of interface of User and Password setting.

4. Tap on the 'Turn IP Camera Server' at the bottom of the screen as shown in figure below. This will activate your front and back camera at the same time and open them on your phone.



Figure 9 screenshot of interface of 'Turn on IP Camera Server'

5. In your web browser, type in the IP address shown in figure below in the url box of a new tab to check that it works. You will see the interface shown in the figure below.

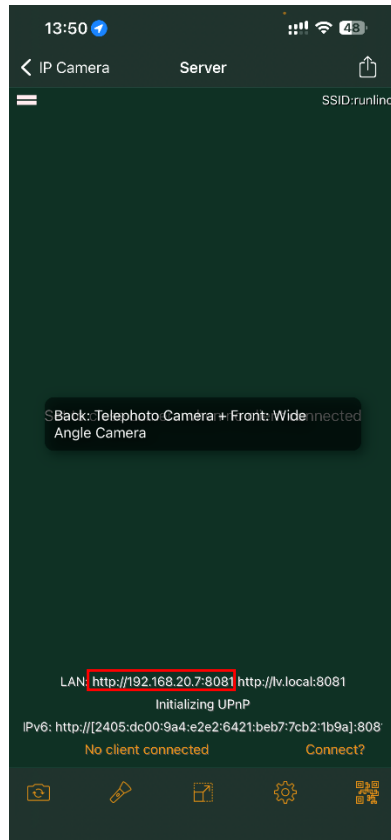


Figure 10 screenshot of interface of IP address.

6. Type in the username and password we set previously in step 3.

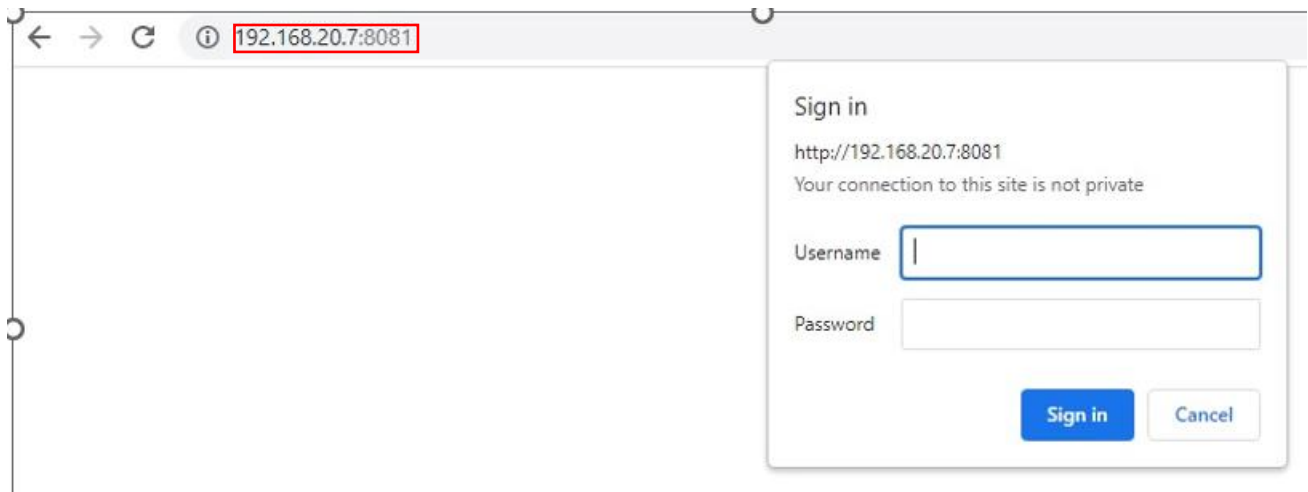


Figure 11 screenshot of interface of signing in

7. Press the button of 'play' and you can see the screen captured by the mobile phone camera can be viewed on the webpage.

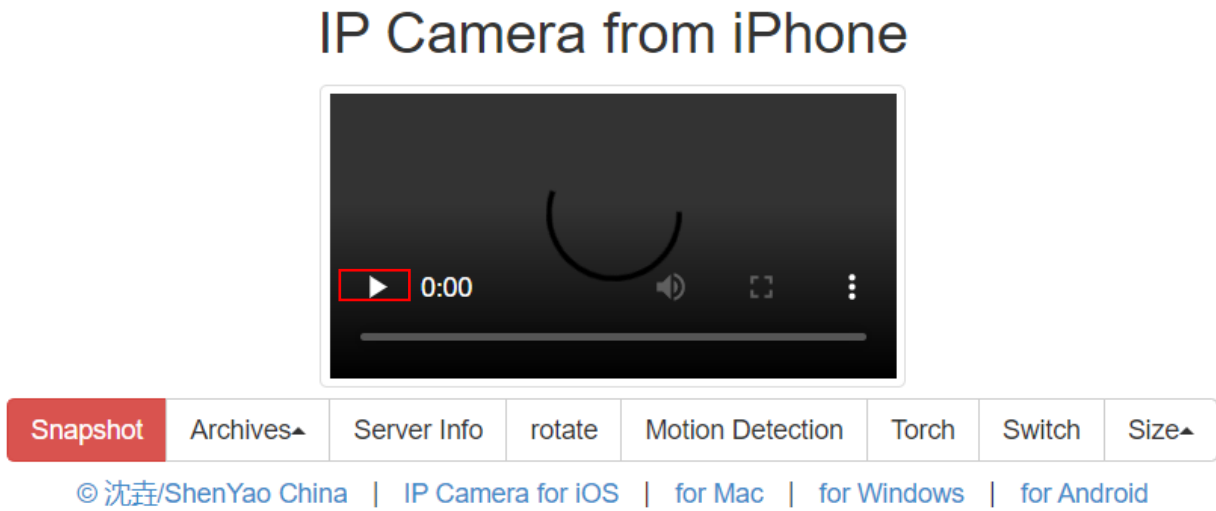


Figure 12 screenshot of interface of screen captured on webpage

Part C2: Program and run the Project (iPhone)

These are the only two lines of code that we will need for this project. One line for a heading, and one line to display our camera feed. This needs to go in the HTML box as shown in figure. Replace the asterisks with the IP address we got in the previous step(`http://192.168.20.9:8081/`).

```
<h2 style="text-align: center;">Robot Camera View</h2>
<iframe style="position: absolute; height: 100%; width:100%; border: none"
src="http://****.8888/"></iframe>
```

The first line will add a heading to your page, it will display 'Robot Camera View' in bold. The second will create a container which displays the camera feed. The 'src' attribute tells the iframe where the content is coming from, so make sure that URL is replaced with your IP address from figure 10 (using the IP shown in figure 10 the URL would be: `http://192.168.20.9:8081/`).

The screenshot shows the runlinc online web interface. At the top left is the 'runlinc online' logo. Below it are two panels: 'Local' with a 'Load File' input and 'Save' button, and 'runlinc.com' with an 'Upload' button. Below these are 'Run Code' (green), 'Stop Code', and 'Board IP' (http://runlinc.com) buttons. A 'STEMSEL' dropdown menu is visible. On the right, there are 'CSS' and 'HTML' editor boxes. The 'HTML' box contains the code from Figure 12, with the IP address replaced by '192.168.20.9'. At the bottom left is a table with columns: PORT, CONFIGURATION, NAME, and STATUS.

PORT	CONFIGURATION	NAME	STATUS
A3	DISABLED		
B4	DISABLED		
B6	DISABE FD		

Figure 13 HTML in runlinc online

Run your code with the 'run code' button in the top left of your page. You should see the camera feed appear underneath the code boxes, like shown in figure15

Note: If you cannot see the camera feed show up, check the instructions that written before the introduction in Page 1. Please check the board IP starts with http:// not https://, as shown in the figure below.

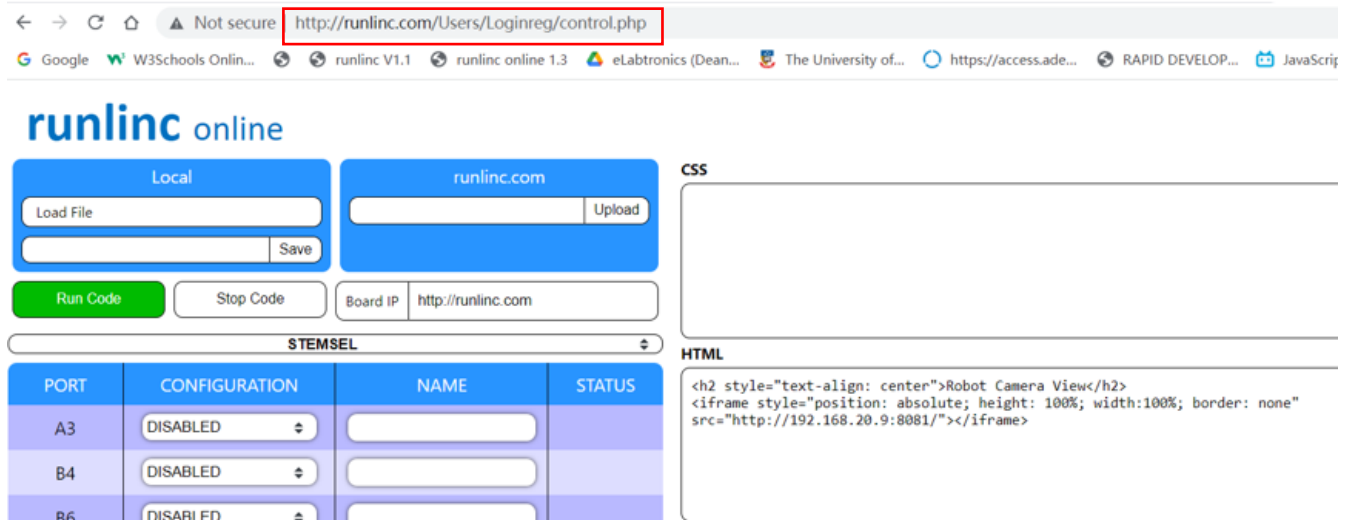


Figure 14 screenshot of interface of board IP of runlinc online

Robot Camera View

IP Camera from iPhone

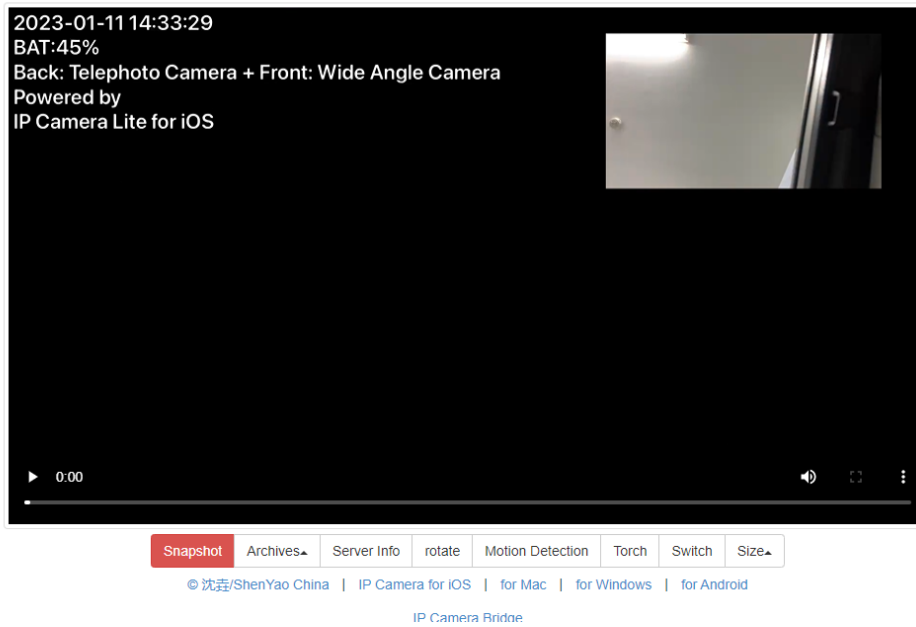


Figure 15 screenshot of interface of screen captured on runlinc webpage