

INTERACTIVE MODEL FOR THE RAILROAD MUSEUM

JULY 18, 2014

Nicole Wong
NICT — Osaka, Japan

RASPBERRY PI

Two formats to streaming

1. RTMP

- Able to stream in to websites such as Ustream
- However, lag is more than 10 sec

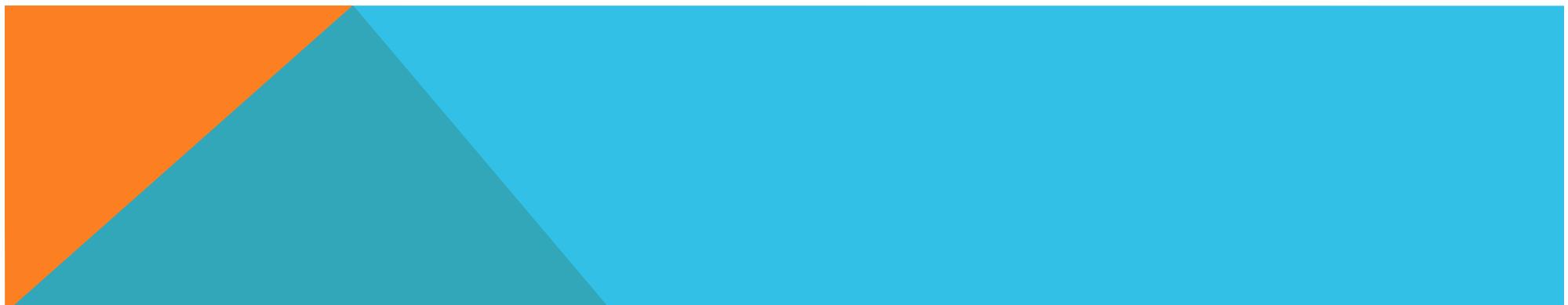
2. MJPG

- Lag time is acceptable, but not the quality
- Proposed format for collaboration with Shimojo sensei's student

Collaboration with NICT and Osaka University

1. Stream MJPG to server

2. Server distribute different quality data depending on the device (save battery)



RASPBERRY PI

Dimension: 8.6 cm x 5.4 cm x 1.7 cm (without battery or any accessories)

~11 cm x 5.4 cm x 1.7 cm (with SD card and Wi-Fi)

~15 cm x 5.4 cm x 7 cm (with small external battery)

Battery life: (support Wi-Fi, camera module, and CPU)

Researches show that 4000 mAh battery lasts ~7.5 hours with Wi-Fi and keyboard

2500 mAh – smaller, but not as long-lasting.

4000 mAh – decent size and last around 2/3 of a day.

5000 mAh and higher – too large and heavy to be put on a moving train.

(Needed to be further tested.)



RASPBERRY PI

After experimenting with various methods:

B E F O R E

Set up using Raspbian OS,
capture using rapistill,
and stream using
continuous photos taken
with rapistill.

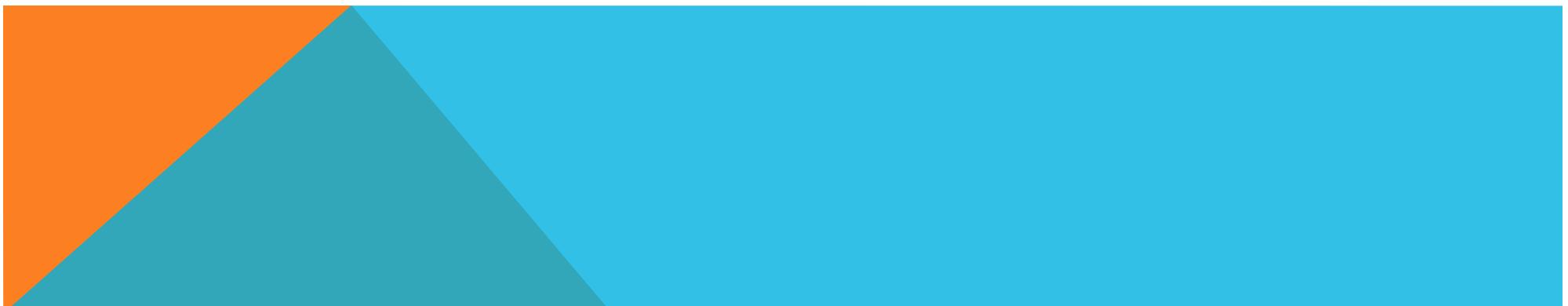
Result: discontinuous, grainy
video.

A F T E R

Set up using Raspbian OS, capture
using raspicam plugin, and stream
as a webcam.
(15 fps 640 x 480)

-different library and plugin

Result: great quality video with
unnoticeable lag(<.5 s)



ETHERNET

Since BPOC server wasn't set up yet, I decided to use Amazon Web Server

Progress:

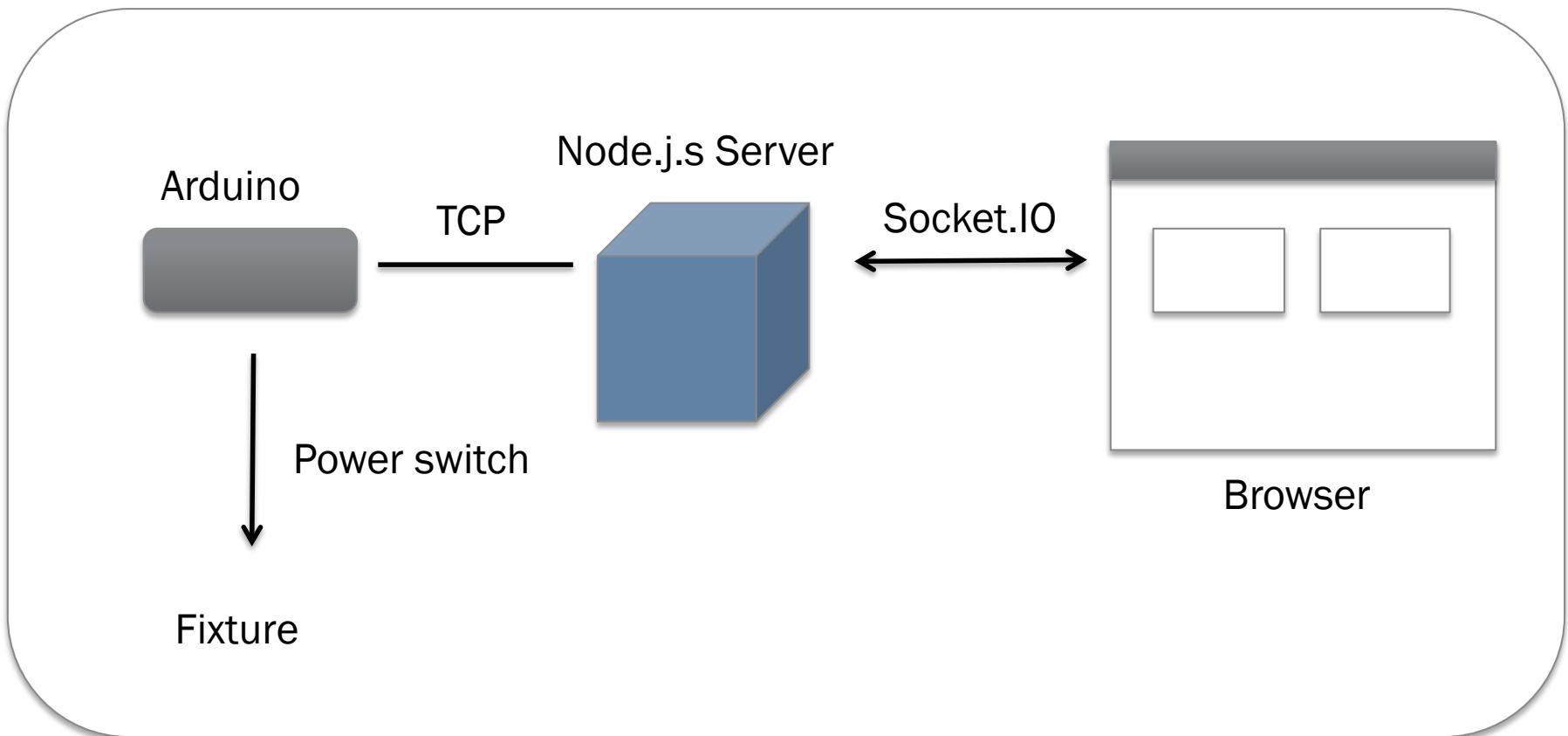
1. Set up website using Amazon S3

- Arduino was able to retrieve data from the static website
- However, it wasn't able to do anything with the JavaScript; only process HTML
- Need a way to store the data

2. Set up server using Amazon EC2

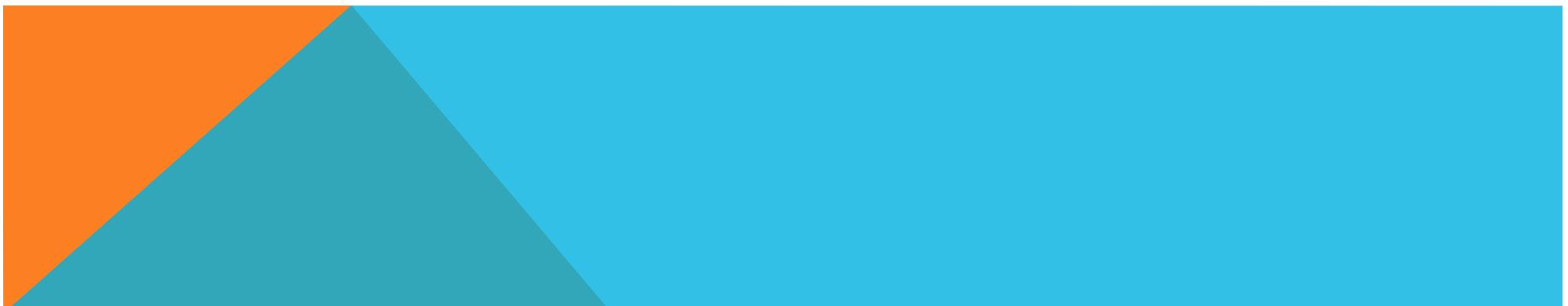
- Created a virtual server Instance running Linux (t2.micro)
- Installed Node.js
- Attempted to create connection between Arduino and web server
- Need to connect web server to browser
- Getting familiar with Amazon EC2 environment





TO IMPLEMENT

- The Node.js server
- Use Node.js to connect to Arduino through Serial Port module.
 - (change it so it connects through Ethernet shield)
- Use SocketIO to send and receive data from Internet
- Use TCP to connect Arduino to web server
- MP3 shield (soldering and coding)



Gion Festival!



Tezuka Osamu Museum
(aka Astro Boy museum)

Minoo Waterfall

ACKNOWLEDGEMENTS

University of California, San Diego (UCSD)

Dr. Gabriele Wienhausen

Teri Simas

National Institute of Information and Technology (NICT)

Dr. Shinji Shimojo

Takata Tomoaki

National Institute of Advanced Industrial Science and Technology (AIST)

Dr. Jason Haga

Balboa Park Online Collaborative (BPOC)

Chad Weinard

