

Magic Smoke Playbook

2018. RTFM

These statements have not been evaluated by the FDA.
Not intended to treat, cure, or prevent any disease.

This Page Unintentionally Left Blank

Contents

2018 - Show Reference.....	4
Hardware Overview	5
First Time Setup	7
Detailed Show Server Setup and Teardown.....	8
Show Audio	9
Show Clock	10
Getting IDs from New Photons.....	11
How to Configure a Show	12
Show Day Procedure.....	14
Fault Protection.....	15
Things to Know	16
Making a Mockery of the Court	17
Safety	18

This page isn't blank.

2018 - Show Reference

Firmware Version

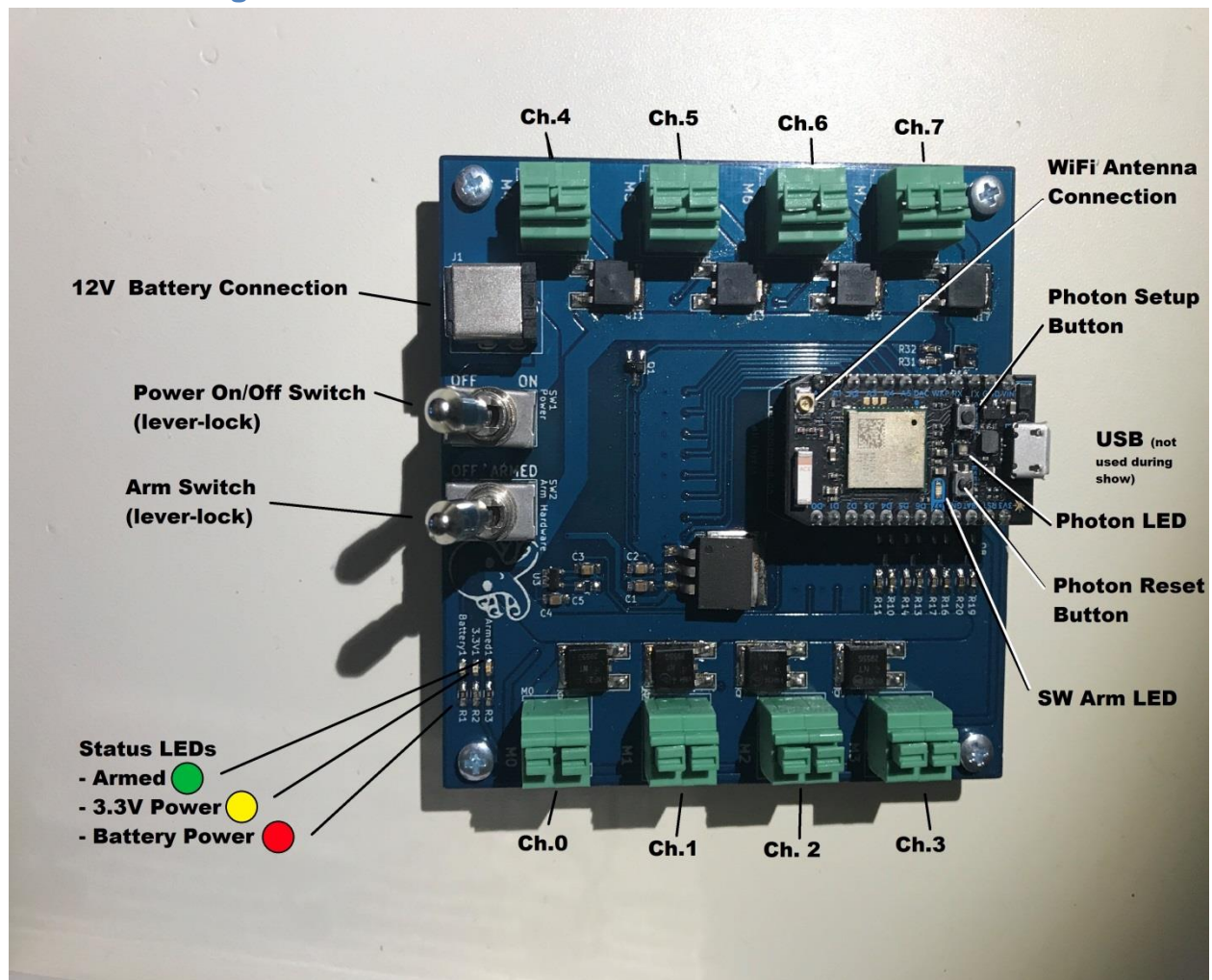
2.0

Board Names and IDs (x35)

- **Alternative Facts:** 2e0021000f47353136383631
- **Ball Ache:** 350027001447353136383631
- **Chubby Despot:** 41003a000f47353136383631
- **Clenched Ringpiece:** 3b0039000647343339373536
- **Clickbait:** 4d0055000c51353532383735
- **Clit Mouse:** 300042000a47353137323334
- **Council of Worthlessness:**
- **Disingenuous Blog Post:** 1f0033000c47353136383631
- **Dr. Lee's Jubiration:** 3e0026000a47353235303037
- **Drawn-out Transgressions:**
- **Fan Service:** 230040000c47353136383631
- **Fucking Assballs Greg:** 3c002a001847353236343033
- **Going Cyclic:** 35002b000f47353136383631
- **Internal Cavity Search:**
- **Jailhouse Hooch:** 360030000f47353136383631
- **Lipid Labia:** 2f001b001447353136383631
- **Menu of Repression:**
- **Moist Molly:** 1f0032000a47353235303037
- **Moose Tracks:**
- **Not the Bees:** 2d003e001847353236343033
- **Orgy of Cannibalism:** 28001a001047353136383631
- **Pain in the Dick:** 3d0025000347353138383138
- **Panty Famine:**
- **Prodigy of Sin:** 20003b001651353530333533
- **Psychosis on Parade:** 30002b001447353136383631
- **Reciprocity Failure:** 3d0035001347353136383631
- **Resplendent Deuce:** 440029000347343337373738
- **Savage Mistress:** 2e0036001747353236343033
- **Shemale Porn Addiction:** 210043000347343138333038
- **Speculation Gone Viral:**
- **Symphony of Squalor:**
- **Thirst Trap Selfie:** 3b001d000c47343432313031
- **Too Many Chromosomes:**
- **Totes Amazeballs:** 300047001247353236343033
- **Wage Slave:**

Hardware Overview

Annotated Magic Smoke Board



Switches and Buttons

Don't fucking touch anything.

Channel Terminals

For the 2018 batch, these are push-button spring terminals. Push on the top to increase gape, release the top for a tighter experience. Do I have to remind you to stress relieve the match wires?

Battery Connection

12 volts are a good number of volts if you want anything to work.

WiFi Antenna Connection

These should already be connected on the boards, but regardless we need external antennas for this shit to work. There is a chip antenna on the board, but the firmware is configured to exclusively use the external antenna, which won't work well if it isn't connected.

Power Switch

Switches battery power to the 3.3V regulator, which powers the photon and other components (in the absence of USB power regulated by the photon).

Arm Switch

Switches the 12V battery power to the high-power side of the firing channel FETs. Hardware arming should generally be performed after the boards have booted up and connected.

USB Connection

Don't fuck with it. It's not used during the show. Unless something is really fucked. If you're going touch it here, make sure Mozo isn't looking, because he might get jealous and stab you in your own sensitive place.

Photon Setup Button

Don't press it, and definitely don't hold it down. And even more definitely don't hold it down while you hold down the reset button. That's worse than smothering a baby otter.

Photon Reset Button

Surprisingly, this button causes a warm reset of the photon.

Guide to the LEDs

Battery Power LED (red)

Battery power is connected

3.3V Power LED (yellow)

Power switch is ON, 3.3V power is provided to components (photon should be powered)

Armed LED (green)

12V power is provided to firing channel FETs

Photon D7 LED (SW Arm LED)

If lit, magic smoke firmware is software armed.

Photon status LED Magic Decoder Ring:

- Flashing green: Attempting to connect to router.
- Breathing green: Connected to router.
- Solid purple mess for 1 second: Identify command received.
- Blue: Expected when flashing the board?
- Anything else: Give up. No show this year.

First Time Setup

Everyone has a first time, but don't fuck it up.

You need an internet connection for this shit, so make sure you're all set up well ahead of time.

The repository for the show server:

<https://github.com/MagicSmokeFireworks/magic-smoke-cloud>

You will need to clone this repository

```
git clone https://github.com/MagicSmokeFireworks/magic-smoke-cloud.git
```

The show server is a node server, which means you need to install node to use it. We also use a variety of third-party node packages, so you need the node package manager (npm) to install all the garbage dependencies. Something like this should work if we have our package configured correctly (from the base magic-smoke-cloud directory...)

```
npm install
```

We also will use a mongo database to record event data throughout the show. So you need to download that separately (google it ffs) and have the mongod running before you try to start the show server (i.e. run `mongod`). The default port is what we expect – choose whatever data directory you want, it does not need to be seeded at this point as we are only using it for data generated during the show rather than configurations.

```
mongod
```

And that's basically it. Sounds easy but it's actually a dependency dumpster fire and a pain in the dick. If you're reading this, then I didn't get around to fully dockerizing the show server this year, but I will soon™.

The mock server is basically the same shit, except this repo:

<https://github.com/MagicSmokeFireworks/magic-smoke-mock>

Detailed Show Server Setup and Teardown

Show server computer IP

The computer (or virtual machine) that is hosting the show server needs to be accessible to the subnet. If you are executing in a virtual machine, make sure that you bridge the network adaptor so that the virtual machine is assigned its own IP on the network.

For the 2018 show, the boards are configured to expect the show server at a specific local IP and port:

192.168.1.99:8080

The show server is set up to listen on port 8080, so all you need to do is make sure that you get the correct IP for the machine. This IP is within the auto-assign range of the router, but is high enough that it should be available if you request that static IP from the router.

Starting the server

1. Make sure your mongodb server is running (i.e., execute **mongod**)
2. Start the show server: **node main.js**

```
$ node main.js
Connected successfully to mongodb server
Listening on *:8080
a user connected
```

That's it.

Stopping the server

Keep track of the console that is running the show server. The normal process for tearing down the server is to send the keyboard interrupt (CTRL+C). Assuming you are running the server in the foreground of the console this will stop it. Remember to pause the show and software disarm the boards first to preserve the preferred baseline state.

Always kill the show server before killing the mongo server, but when you're done with it you can kill the mongo server in the same way (CTRL+C). At the end of the show we will want to harvest some data from the mongoDB, but it will persist in the data area, so it is okay to kill it and we can get the data later.

If needed, you can also find the "node" and "mongod" processes and kill them directly.

Show Audio

The 2018 show server will stream audio to clients on the show page, playing, pausing, and seeking the audio file to correspond to the show clock. To use an audio file, place either a .wav file named “show.wav” or an MP3 file named “show.mp3” into the magic-smoke-cloud/static/ directory. If you have both a wav and mp3, the wav will be used preferentially.

Because the music is streamed to the client, it will play through the browser on the show page. This means that any client on the show page will get the audio, but there will be slight delays given lag in the network. It is therefore recommended that the show client be connected on the same computer that is running the show server and the audio be connected to that machine.

Additionally, given that the show server is controlling the show clock and the client browser is playing the music file, there are events synchronizing these together that can be disrupted if you close or refresh the page. If you refresh the page while the show clock is ticking, the browser will stop playing music until you “interact” with the page (i.e. click in dead space or something else). This is a browser behavior that we can’t change.

TL;DR There’s no reason to refresh the show page, so don’t refresh the show page.

Show Clock

You asked for it mother fuckers so here it is. You're giving the computer more control, which is brave.

Things to know about the show clock:

- The clock is maintained by the server. What you see on the show page is only a visualization into that master clock. This has several implications... like: just because you don't see the clock ticking doesn't mean it isn't.
- Closing the client/browser will have no effect on a ticking clock. There is no watchdog for client interaction. If you lose controllability of the show server from the client, you can stop the clock ticking by killing the show server.
- The clock ticks at 10Hz, meaning events are handled every 0.1 seconds. The show clients should see each of these ticks come across the network, but if something lags and you don't see a tick that doesn't mean it didn't happen, you just didn't see it.
- While the show clock is running, it will detect groups to fire on the **entry** into a time. This means that if you have a group at time N, it will be fired when the clock ticks from N-0.1 to N. In other words, if you start the clock at time N, then a group at time N will not be fired.
- The clock starts at 0.0 and will not let you move it to negative time. This means that any groups configured with negative time or 0.0 time will never be automatically be fired by the show clock (but can be fired manually).
- Other than starting and pausing the show, there are three ways to adjust the clock:
 - The + button next to the clock will increment the time by 0.1 seconds
 - The – button next to the clock will decrement the time by 0.1 seconds
 - Every group on the show page has a “jump to” button that will jump the show clock to 0.1 seconds **before** the group. Meaning that group will be fired on the next tick.
- These time adjustments can be made while the clock is ticking and will not cause the clock to stop ticking, it will jump without interrupting the overall 10Hz cadence.
- A clock jump will not fire a group. In other words, if the clock time is at N-0.1 and you increment it to N, this will not fire a group at time N. This is the main reason that I recommend you not jump time while the show clock is running. If you increment time forward just right, it is possible to jump a group without firing it.

Getting IDs from New Photons

Background

The 2018 show consists of 35 magic smoke boards (possibly with a spare). Last year there were 25 boards, but some got a little crispy, so ultimately there are perhaps 12 new photons in the set this year. In order for the show server to understand which boards are connected (and send them commands) the server must be able to associate photon IDs with board names. Each photon gets a unique ID number from the factory, which is what we rely on for unique identification during communications. Before we can operate the new boards for the show, we need to assign their IDs to the board names. There are already 35 board names picked out, with the 25 returning from last year still allocated their original ID numbers. The server picks up the assignments from the “boardinfo.json” file every time it starts, so you need to save your changes there.

TL;DR You’re the bouncer at the club. You need to check everyone’s ID and make sure there aren’t any fakes.

Your Mission

(should you choose to accept it, but you do)

1. Start up the server
2. Power on the boards one at a time, repeating the following steps per board:
 - a. Watch the photon LED for indication that it has connected to the WiFi
 - b. Inspect the server console output to see if you get the unknown ID message

```
$ node main.js
Connected successfully to mongodb server
Listening on *:8080
Unknown board id sending status packet: ae0021000f47353136383631
Unknown board id sending status packet: ae0021000f47353136383631
Unknown board id sending status packet: ae0021000f47353136383631
```

- c. If you get an unknown ID:
 - i. Copy that ID from the console
 - ii. Open up the “boardinfo.json” file in the server directory and find a new board (id field starts with “new”) or if you have already assigned all 10 new boards, you have to figure out which old ID is not re-used, so come back to this board.
 - iii. Save the boardinfo.json changes and restart the server and confirm via the “identify” command that the new board is named.
 - iv. Record the board’s name on the board itself via tape or whatever so you don’t get them mixed up.
- d. If you do not get the unknown ID message:
 - i. cycle through the boards, sending identify commands until you get the photon to light up.
 - ii. Record the board name on the board.

How to Configure a Show

Background

A magic smoke show consists of **groups**. In the 2018 show, groups are timed events. Each group you create must have a time and you may optionally give it a description so you can remember later what the fuck you were smoking. The times will be fixed to the nearest 0.1 seconds because the show clock ticks at 10 Hz, so you should keep this in mind when you are creating groups. Groups will appear in the group config page and the show page in time order. If you wish, you may configure multiple groups for the same time, but in that case the ordering is not controlled. At its most basic, a group in the show is just a timed event with a description that you can use as a cue at that time and you can use it to jump the show clock to that time. In other words, there is no requirement that a group have firing channels associated with it – you can use groups as organizational structures or useful time jump points. However, the primary purpose for a group is as a vessel to associate one or more board channels with a firing time. Each of the eight channels per board can be configured to be linked to a single group. Each channel can also optionally be given a description of what specific effects are tied to that channel. The channels linked to a group will appear in that groups place in the show page and all channels per group will be fired together either when the show clock ticks into that group time or the operator clicks to the fire button for that group.

Configuring Groups

- Access the group config page from the parent config page
- You can add groups using the panel on the left. You must specify a time for each group you add. A description is recommended but optional.
- You may have multiple groups at the same time, if that's something you think is cool.
- Each group is assigned a random, unique ID number which is used to keep track of channel assignments. Therefore you may change the time and/or description of a group at any time without disrupting the channel assignments.
- If you want to edit a group time or description, do so from the group config page, but only edit one group at a time. Be sure to save or cancel your edit before editing or deleting another. If you attempt to edit more than one, your edits may get lost.
- You can also delete unwanted groups from the config page, but murder is immoral so you should really consider putting them up for adoption instead.

Note: If channels are configured to a group that you later delete, those channels will not be part of the show unless you reconfigure them to an existing group.

Configuring Boards

- Each board has its own configuration page
- You can access this from the parent config page, from the boards page, or from an individual board page
- If you are struggling to reach the lower board names using the navigation lists at the top of the page, visit the “boards” page by clicking on “boards” directly. This will show you a page that contains links to all individual board and board config pages directly.
- If you want a board to be in the show, you should specify a location. Preferably a useful location that you can reference later, but anything other than “inactive” will have the board appear in the status page as active.
- Configure boards one at a time, and remember to hit the “save” button at the bottom of the page to save your changes. You can change as many fields on a single board as you want before saving, but just remember to save the changes or they will go away.

Show Day Procedure

Show Prep

- Get the boards, router, and server computer all running on battery. Laptop computer can be on AC as long as it will seamlessly switch to battery on power failure.
- Run AC power for the speakers, radio transmitter, and anything else that is not in the communication path for the show.
- Tape. Down. Cable. Runs. FFS.
- Confirm that the speakers will play from the computer that will be running the show page (the client gets the audio).

Show Imminent

- Field clear of non-essentials
- Router ON
- Server start (mongo and show server)
- Audio connected/speakers on
- For each board in field:
 - HW Arm OFF (should already be OFF)
 - Power OFF (should already be OFF)
 - Battery connected
 - Power ON (watch for connection: breathing green)
- At server as boards come online:
 - Look at status or show page, confirm all expected boards connect
 - Check resistance values for configured channels, if not performed manually
 - Investigate any reported signal quality issues
 - Investigate any reported match resistance issues, if necessary

Commit to Show

- Field clear of anyone not arming boards
- For each board in field:
 - Confirm that SW Arm LED is not lit
 - Switch the HW Arm switch
- At server:
 - On show page, confirm that boards are SW disarmed, but HW armed

Start Show

- Field clear of everyone
- Software Arm ALL
- Confirm all reported armed
- At desired T-0.0: **Start Show**

Fault Protection

Hint: it's you!

Safe Response

While the show is running, you should have the primary client on the show page at all times. Any deviations away from the show page or side investigations should be performed while the show is paused or from secondary client. The best way to pause or stop the show quickly and cleanly is to have this show page in good connection with the server and command a stop of the show clock (as well as a software disarm of all boards if there is a board safety concern).

However, always be prepared to kill the show server. In the 2018 show, the server is driving the show timing, and not having the client viewing the show page does not stop a running show from continuing. Killing the show server is not the cleanest way to pause or stop a running show, but in cases where interaction between your client and the server is questionable, it should always be safe to kill the server to ensure that firing commands won't be sent out. However, if you have a good connection with the server, it is preferred to pause or stop the show by stopping the show clock and, if desired, software disarming all the boards. This is the cleaner and safer way to pause the show, leaving all elements in a known and easily recoverable state.

Fault Detection

The show server runs an open-loop control. No feedback from the boards is considered in the process of ticking the show clock and sending fire commands. This makes the behavior simple and predictable, but means that you should be aware of the state of the system. If a board resets for example (not anticipated to occur) it will go offline while it reboots and reconnects and eventually come back up in a software disarmed state. If you detect this condition, you should send a software arm command (software arm all from the show page is fine) in order for this board to fire channels. Something like this is fine while the show is running, but any more significant issues you may need to pause the show to diagnose.

The show page is intended to provide the high level information you need to identify major faults in the state of the system. The status on the side panel will alert you to connection or arming issues and the individual badges for the channels in groups will provide status on match state, firing state, and overall ready state. If you intend to detect issues during the show, you should familiarize yourself with the information available in these badges and review it prior to the show to identify any unexpected deviations while the show is going.

Additionally, the show clock will not repeat a fire command that may have failed, but if you detect this condition during the show, and you want to try to get the missed effects in late, you can manually click the fire button for any group.

Things to Know

Fun facts! There will be a quiz at the end. If you get anything wrong, Mozo will have his way with you. (Don't look so excited)

- The show is open-loop control. Group fire actions are deterministic and timer driven. When the show clock ticks into a group time, fire requests are made for those channels. No feedback from the boards is considered before or during this time. Even if the show server is aware that a board is disarmed, disconnected, or has reset, it will not command anything other than a fire. If the show server detects that a board has not received or responded to a fire command or that a match appears to not have burned, it will not attempt a re-fire of those channels. All of this higher-level of awareness is bestowed upon ye of human flesh. It is possibly safer and certainly less fragile this way. We have avenues to close the control loop in the future if we so dare.
- If you are running the show and want to quickly stop, the best approach is to click the "pause show" button. If you have any doubts about your connection to the server, you can always kill the server as well.
- It is perfectly fine to define a group with no assigned channels (and in fact may be normal to do so to identify good time jump spots or show cues).
- It is also perfectly fine to define multiple groups at the same time (not sure why this would be good though).
- If you see strange feedback that you want to investigate during the show, either pause the show or use a secondary client to dig into the telemetry. You should not navigate away from the show page while the show is running (for a lot of reasons).
- Stress relieve your cables for fucks sake
- We are using mongoDB this year to record all events (known to the server) that occur during the show. This includes command requests received from the client arm/disarm/fire, group fire requests made by the show clock, and all telemetry received from the boards. All time stamped for our reconstruction delight after the show. In future years, we will likely have all the configurations in here as well and will allow the show server to automatically recover from errors and other fancy shit.
- Speaking of fancy shit, did you know that docker doesn't play nice on windows 7? I would have dockerized this year and made the need for most of this documentation obsolete, but I gave up because how much of a pain in the dick it was to do everything through the virtual machine. Next year though for sure...

Making a Mockery of the Court

For the purposes of testing and playing with the server, there is a tool (magic-smoke-mock) that you can use to emulate a magic smoke board on the network. The mock will implement an HTTP server and a NET server on two ports of your choosing, the HTTP server serving a page that you can use to view and control the mock behavior and the NET port emulating the interactions that a board has with the server.

From the perspective of the show server, it does not strictly know that a mock server is different from a real board, except that it is not sitting at the default port and is on the same host.

At the moment, the mock NET server assumes that the show server is on the localhost, rather than the static IP, but this can be easily changed if you want to exercise across a network.

There is a readme in the mock repo, but basically you run it like so:

```
node mock.js <boardsname> <httpport> <netport>
```

Where the boardsname is the short name for a board (e.g., af for Alternative Facts), the http port is where it will serve the control page, and the net port is where it will interact with the server. The ports don't matter as long as they don't conflict with other servers. 8081 and above is good territory.

If you start more than one mock, give them all unique port numbers.

Running a mock board and that actual board at the same time can give you strange behavior as the mock board may mask the IP of the actual board and you may see commands going to one or the other depending on timing of latest telemetry packets. This can also obfuscate connection issues observed with the actual board. Therefore you should not run mocks on the server that you bring up for the actual show.

Safety

A few important safety reminders



FIGHT
the **BITE**

