Open Lighting Architecture: Blinky Lights!

Matt Ranostay
matt.ranostay@intel.com

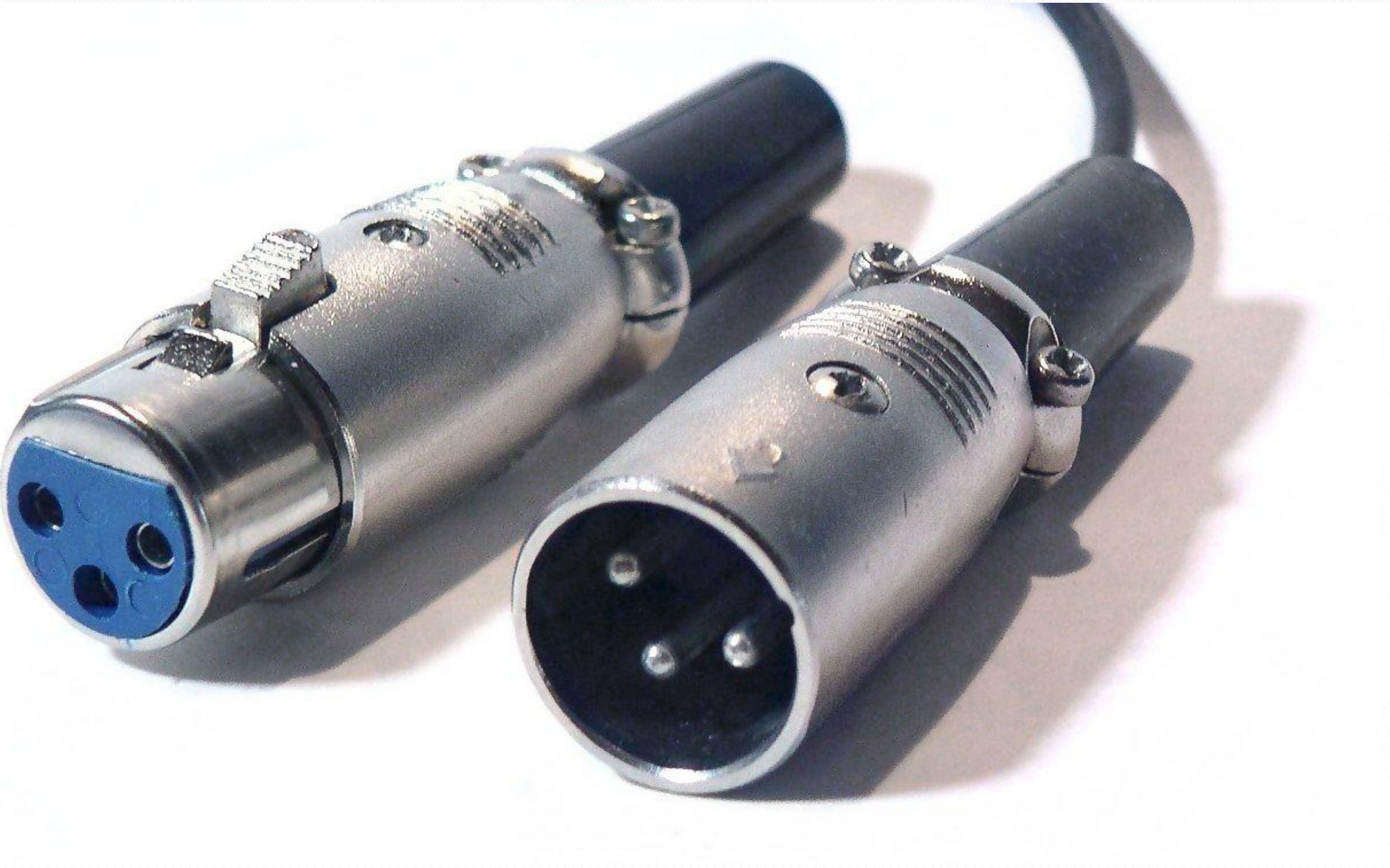
Brief Overview

- Crash course on lighting systems, and how HW + SW interact to drive lighting setups for both commercial, and simply hobbyists.
- Demoing how **open source** can open a previously very proprietary, and expensive to implement real world lighting setup.
- Not my day job, clearly, and don't claim to be an expert in industrial lighting systems.

DMX? Not just a rapper?

- DMX512 (Digital Multiplex 512) is an industry standard lighting specification that has been in use since 1986 (literally as old as me)
  - Any light show, rave, concert, and misc events you have been too has certainly used this technology
  - RS-485 is the standard used for data transmission
  - XLR-3 or XLR-5 (typical) connector
    - XLR-5 has two unused or unassigned pins
  - Cat 5 or 6 cable is also common

XLR 3-Pin Plug



DMX RS-485 Traffic

- DMX512/E1.11 is named so because it is 512 slots of data + 1 start code slot
    - Slot is one byte
    - Each of these 512 slots is called an universe
- Specification allows up to 44 fps, but can be piped out faster if not all slots are used.
    - Most vendors stick to 44 fps though
- Slot data can control panning, light intensity, etc (i.e. command + data slots)
- Slots can be combined to increase range from 8-bits to 16, 32, etc bits.

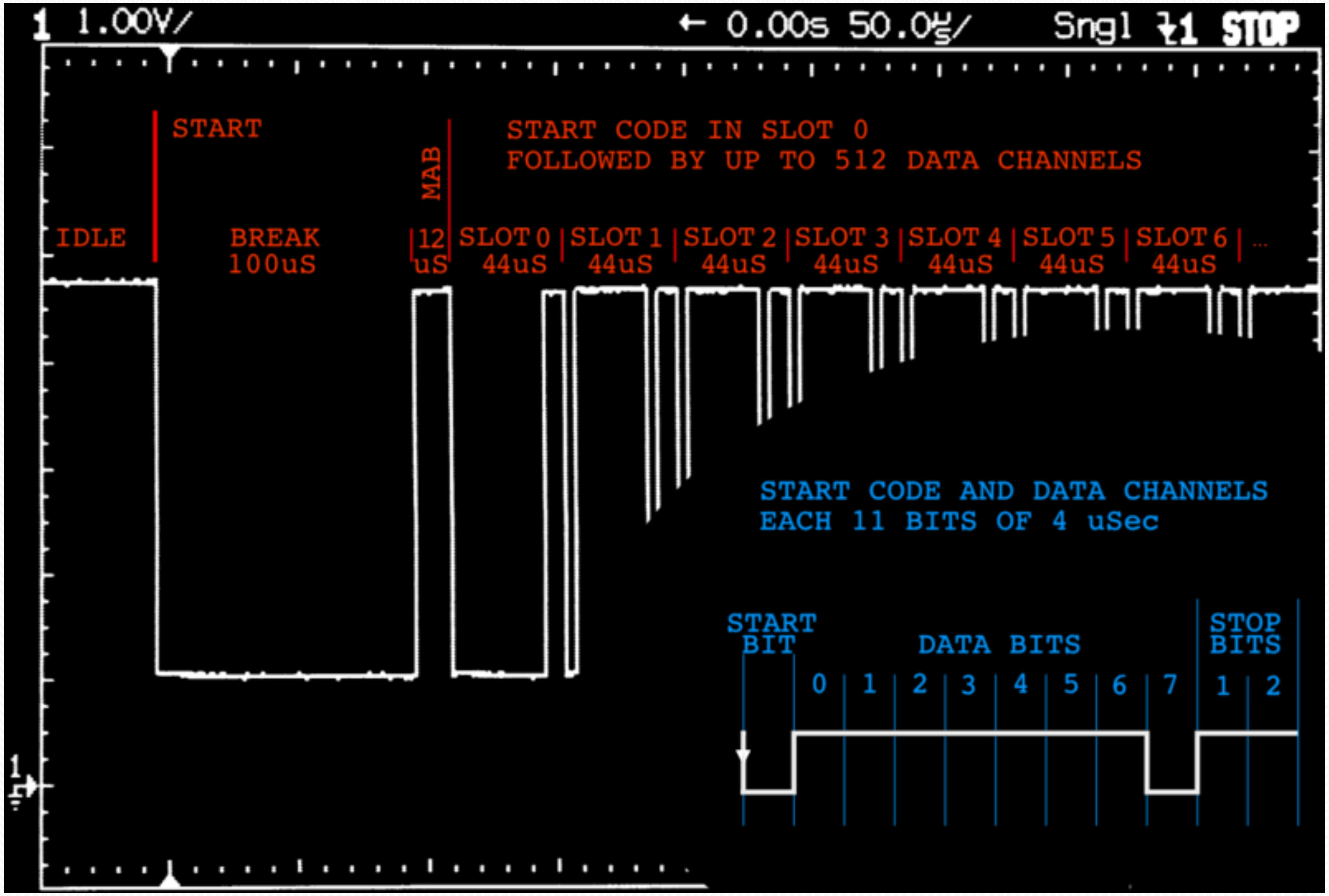
DMX Device Interaction

- DIP switches or dials set the start slot of the device
  - Non-discoverable busses onboard
- Each is a one load unit typically on a RS-485 bus
- Devices ignore slots that aren’t assigned to them or start codes they don’t understand
- Most devices respond to a the NULL start code
  - Text packet, system information, and proprietary vendor codes

DMX512 RS-485 Protocol

- RS-485 has no clock line so you need a set way to know when a frame begins
- Must be terminated for the bias network to work (120 ohm)
- 32 unit loads is the maximum allowed
- DMX512 signals a start of a frame
  - Break of 100 microseconds
  - Mark After Break (MAB) 12 microseconds
  - Followed with 11-bit sequence for each slots
    - Start bit + 8 bits of data + 2 stop bits.

DMX512 Oscilloscope Output



Example of DMX-enabled Devices

- Stage Lighting (vast majority of applications)
  - Lasers
  - RGB LED panels (i.e. 16x16 grids)
  - RGB LED strings
  - Servo controlled light mounts (Pan + Tilt)
- Dimmers
- Haze / Fog Machines
- Niche market of animatronics
  - Halloween decorations

DMX in action (Singapore Flyer)

DMX in action (Example Setup)

- Universe #1 – 50 RGB LED string
  - Slots 1 – 150
- Universe #2 – #1 half of a 256 RGB LED panel
  - Slots 1 – 384
- Universe #3 – #2 half of a 256 RGB LED panel
  - Slots 1 – 384
  - Latching output of panel
- Repeat for other hootenanny LED panels and strings action

DMX-over-IP

- Various implementations over the years that have all have pros and cons
  - DMX specification doesn’t reference any networking protocols, but all of the wire protocols mirror the RS-485 data transport
- Typically Ethernet is used as the physical link layer
  - Wi-fi and other RF based physical layer transmissions methods exist

DMX IP/UDP Transport Methods

- ART-Net III
  - More of a hobbyist implementation
  - 32,758 universes allowed per network instance
- E1.31 – Streaming ACN
  - De facto solution used for large lighting shows
  - 63,999 universes allowed network
  - ANSI + PLASA/ESTA approved standard
- Operate in both unicast or multicast mode

Remote Device Management

- RDM/E1.20 was designed after DMX512 and uses the same RS485 path to send messages
  - Bi-directional control and status messages
  - Backwards compatible with legacy DMX devices
  - Uses start code 0xCC and encapsulates RDM data within this
  - Discovery of devices is a binary search of UIDs (2 byte ETSA/PLASA vendor and 4 byte device id)
- E1.33/RDMnet under development/spec review
- Passing OOB messages over the RS-485 link

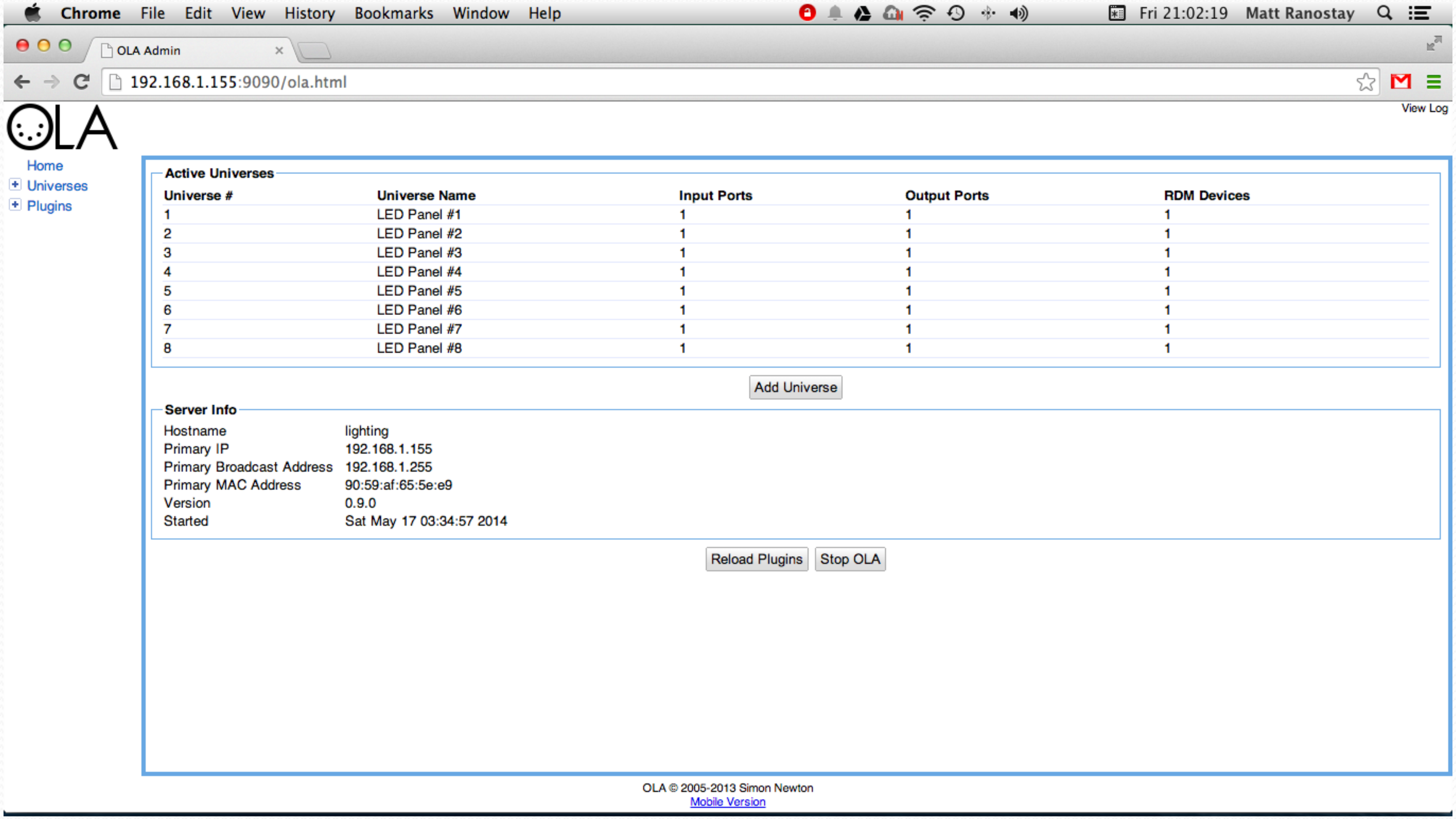
Open Lighting Architecture

- Completely open source implementation of the networking aspects of the DMX512 protocol
  - OLA loopback protocol for API bindings
  - Open Sound Control (OSC)
  - ArtNet III + E1.31 sACN
- Various plug-ins for interfacing to devices
  - Vendor specific plugins
  - SPI (SPIDev)
  - USB DMX devices
  - RS232 UART to RS485

Open Lighting Architecture

- Future support for RS-485 array of devices on Linux is planned which will make this more of a network to DMX512 bridge
  - Cheap low end SBC then can be a DMX bridge
  - Currently SBCs have no RS-485 native connection so RS-232 to RS-485 bitstream translation is required
- Remote Device Management (RDM) is fully supported and robust
  - Lighting control panels or various control devices

OLA Web Interface


Chrome File Edit View History Bookmarks Window Help
Fri 21:02:19 Matt Ranostay
OLA Admin
192.168.1.155:9090/ola.html
View Log
OLA
Home
Universes
Plugins

Active Universes

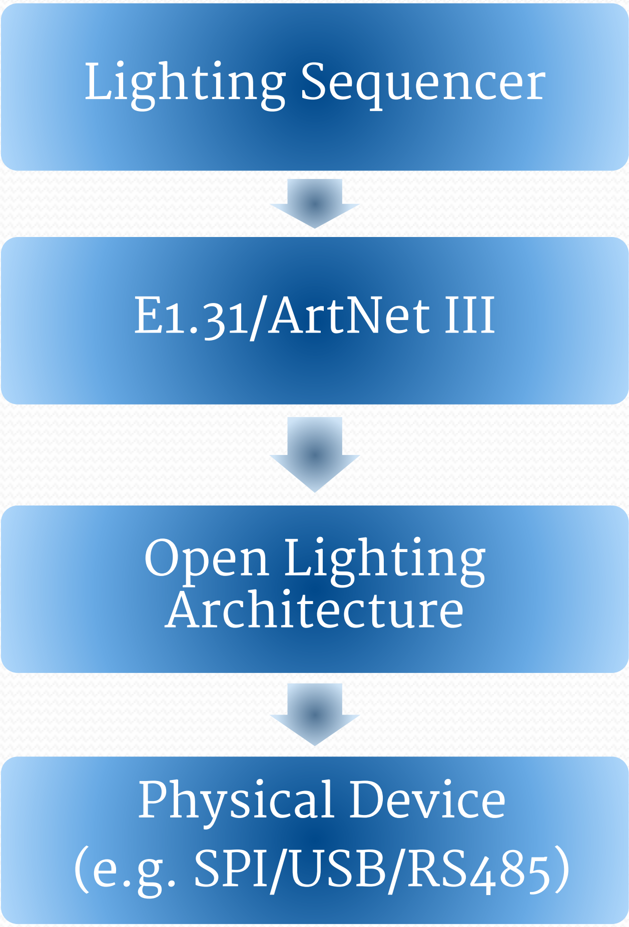
Universe #	Universe Name	Input Ports	Output Ports	RDM Devices
1	LED Panel #1	1	1	1
2	LED Panel #2	1	1	1
3	LED Panel #3	1	1	1
4	LED Panel #4	1	1	1
5	LED Panel #5	1	1	1
6	LED Panel #6	1	1	1
7	LED Panel #7	1	1	1
8	LED Panel #8	1	1	1

Server Info

Hostname	lighting
Primary IP	192.168.1.155
Primary Broadcast Address	192.168.1.255
Primary MAC Address	90:59:af:65:5e:e9
Version	0.9.0
Started	Sat May 17 03:34:57 2014

OLA © 2005-2013 Simon Newton
Mobile Version

Typical OLA Setup



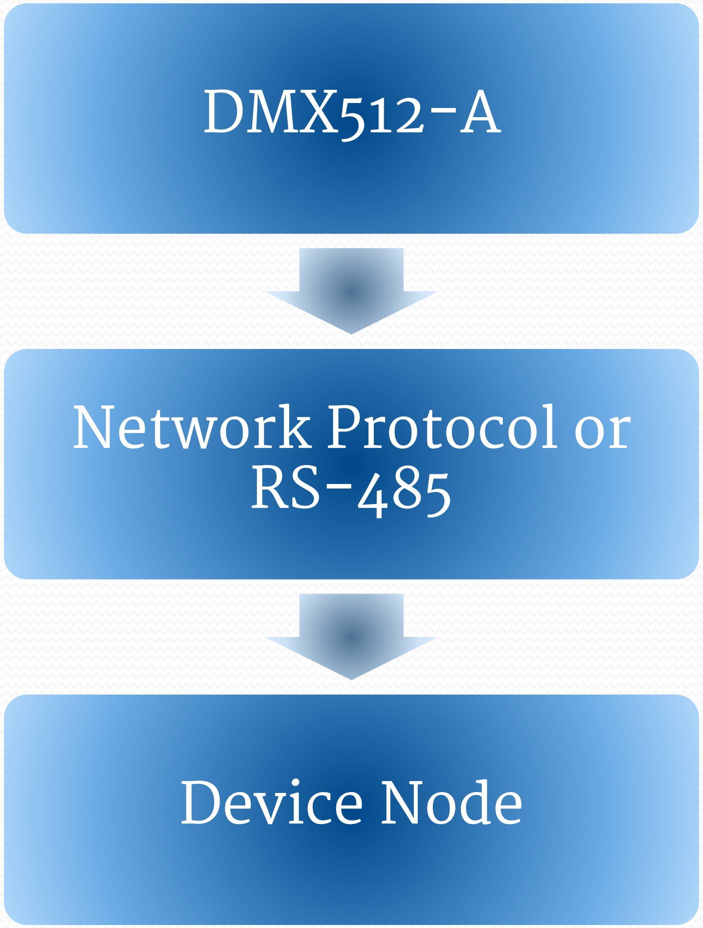
Lighting Sequencer

E1.31/ArtNet III

Open Lighting Architecture

Physical Device
(e.g. SPI/USB/RS485)

Data Flow



DMX512-A

Network Protocol or RS-485

Device Node

OLA Nonsense Hacks

- Triggers from DMX control codes that run commands
  - Could trigger a processing script
  - Toggle a relay switch
- Python/C++/Java loopback API allows you to send frames locally to a OLA interface
- JSON web interface for when other protocols are too much overhead, or for interfacing with an web application

OLA JSON Endpoints

- JSON/RESTful interface
  - /set_dmx
    - POST u=[universe], d=[DMX data (a comma separated list of values)]
  - /get_dmx
    - GET u=[universe]

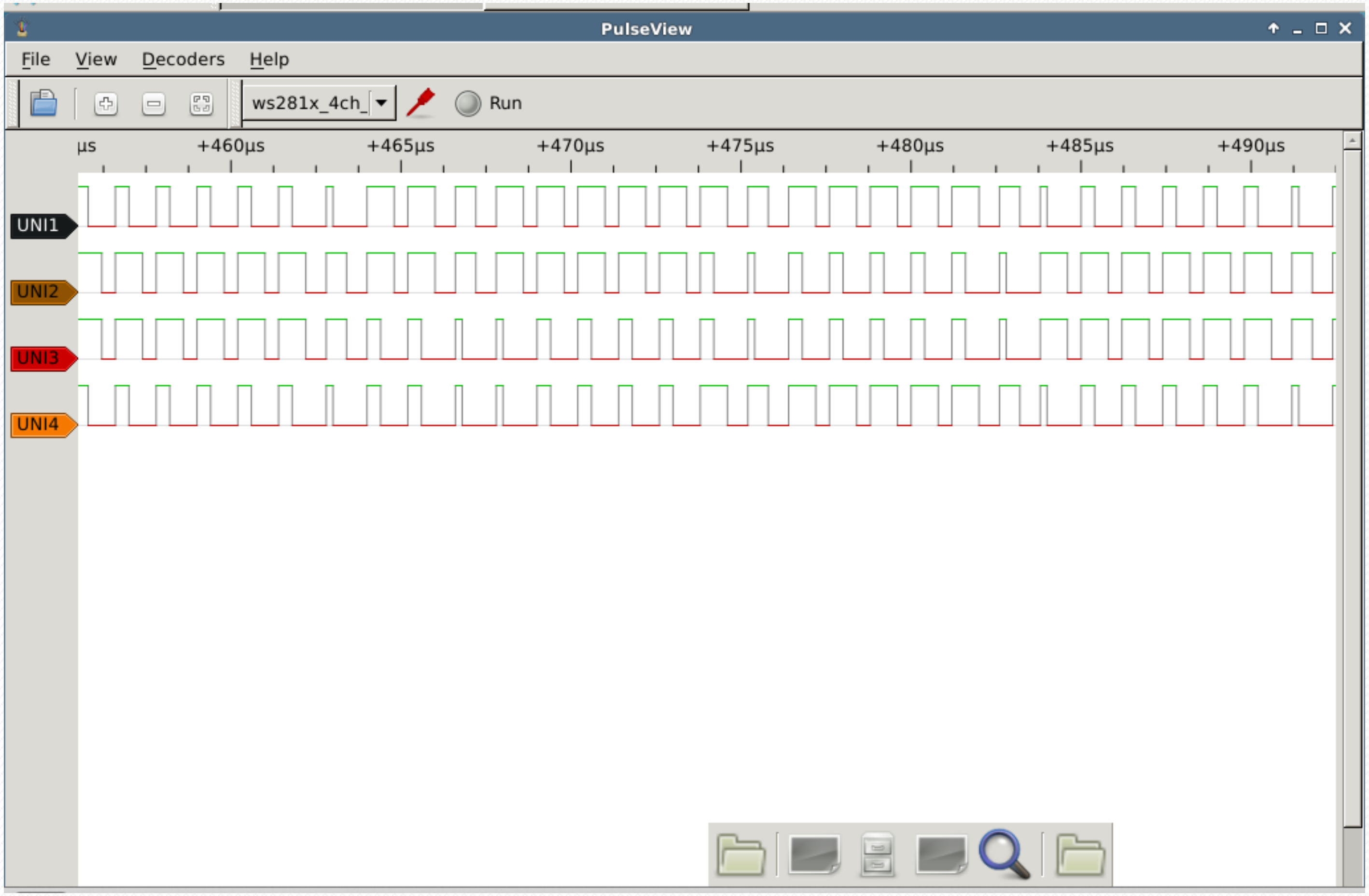
Open Sound Control

- OSC is used for MIDI control as well as lighting control
- OLA has a OSC interface that can be used as well
- Various DMX frontend generators use this for user input
  - Open Lighting Architecture
  - PixelController
- TouchOSC – iPhone application to OSC systems
  - Several applications exist for Android and iOS

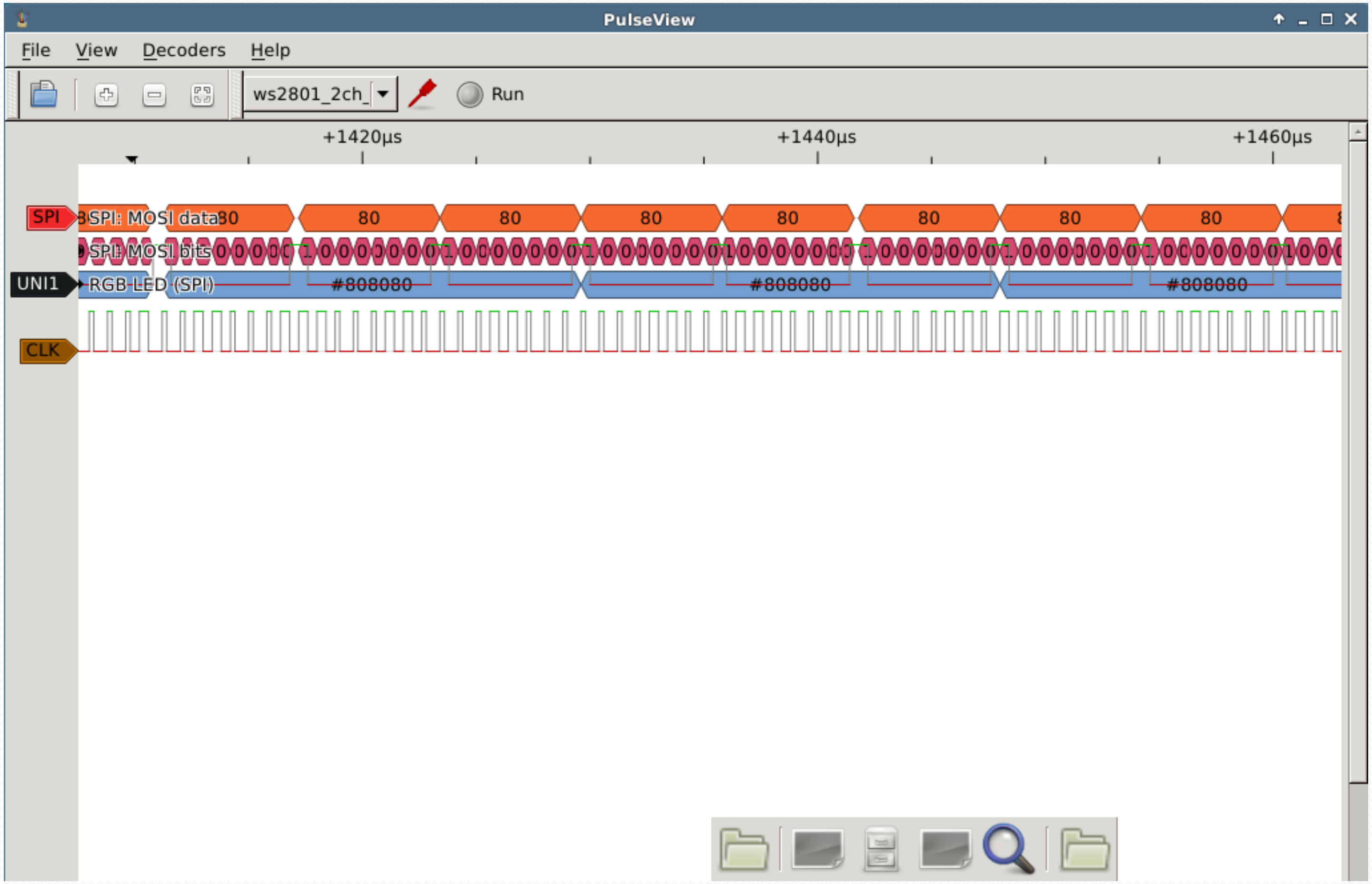
Common LED Types

- WS2801 “SPI-like”
  - SCLK and MOSI/DATA lines
  - Latching is done by 500 microseconds low signal on the clock line
- WS281x
  - Self-clocking over one-wire
    - Logic high - ~2.5 uS high
    - Logic low - ~1.25 uS high + ~1.25 uS low
    - Latch 50 uS
- Patent troll has the RGB PWM rights (and lot of Chinese LED driving ICs violate these)

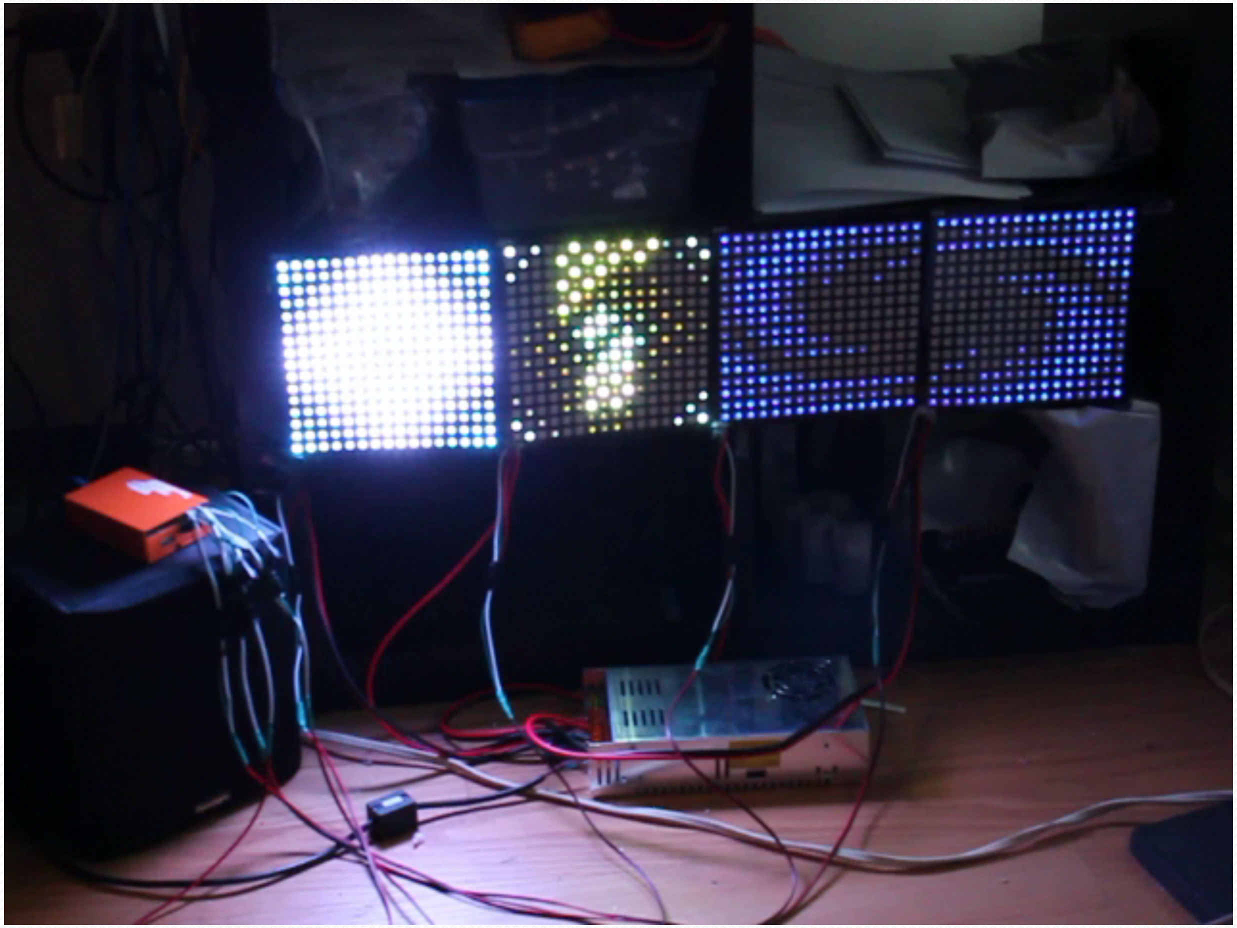
WS2812 Chipset (self-clocking)



WS2801 (SPI + protocol decode)



OLA + PixelController + AM335x PRU



When Determinism Matters

- WS281x is an example you can’t use pure SPI or bitbang from userspace and expect it to work
- BeagleBone Black (am335x) + PRU (Programmable Real-time Unit)
  - OLA SPI plugin interacting with faux-spidev interface
- Any microcontroller/FPGA/CPLD connected to a interface controllable can be used
  - Input should be SPI so it is transparent to OLA

Conclusions/Opinions

- DMX is “legacy” but it is a classic and has little need to be improved other than legacy systems being moved to the networking space versus RS-485 transport
- Inferior protocols exist for the sheer fact that people are too lazy to learn something new, and figure it is okay to write a new one
  - Open Pixel Controller is an example

Demo

- MinnowBoard Max
  - SPI + WS2801 LEDs
  - OLA SPIdev plugin
  - DMX End Node
- Setting up OLA + configuring a LED string universe
- JSON/RESTful API + Python API + Web Console interface

Questions

- How many of you have used stage lighting that has utilized DMX512? Or some other proprietary protocol?
- Have you used OLA, and how can we improve the project?
- What are your currently using for your software driven lighting needs? Or what you do use for Xmas lighting?

Special Thanks

- Simon Newton – nomis52
- Andrew Frazier – mrpackethead
- Pantelis Antoniou – panto

References

- Open Lighting Architecture
  - http://www.openlighting.org
  - http://rdm.openlighting.org
- DMX512 Oscilloscope Graph
  - http://en.wikipedia.org/wiki/File:Annotated_trace_of_DMX-512_signal.png
- DMX512 Wikipedia
  - http://en.wikipedia.org/wiki/DMX512
- DMX Start Code
  - http://www.erg.abdn.ac.uk/users/gorry/eg3576/start-codes.html

References

- Sigrok + Pulseview (WS2801 + WS2812 debug)
  - http://sigrok.org
- WS28xx PRU firmware
  - https://github.com/mranostay/ws28xx-lighting-pru
- PixelController
  - https://github.com/neophob/PixelController
- TouchOSC
  - http://hexler.net/software/touchosc

References

- WS2801 Datasheet
  - http://www.adafruit.com/datasheets/WS2801.pdf
- WS281x Datasheet
  - http://www.adafruit.com/datasheets/WS2812B.pdf
- Philips PWM RGB Patent
  - http://www.colorkinetics.com/technologies/ipinfo/patents/pdf/US-6806659.pdf
- Singapore Flyer Image
  - http://commons.wikimedia.org/wiki/File:The_Singapore_Flyer_from_Marina_Barrage_(8050344592).jpg