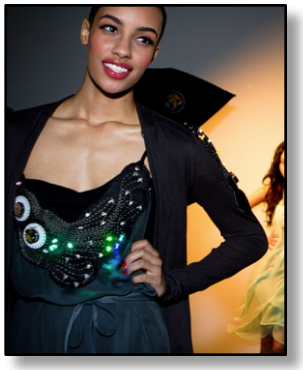
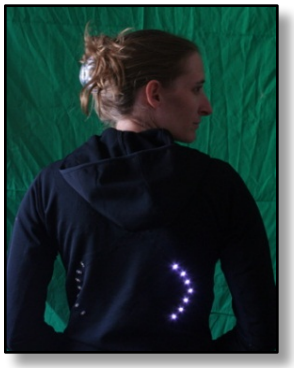
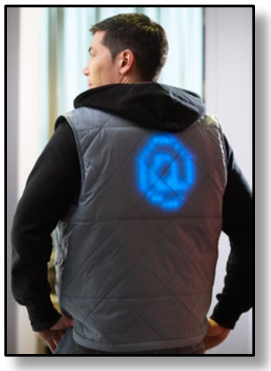



WiFi-enabled LED Workshop

Rui (Ray) Wang

CICS

UMass Amherst

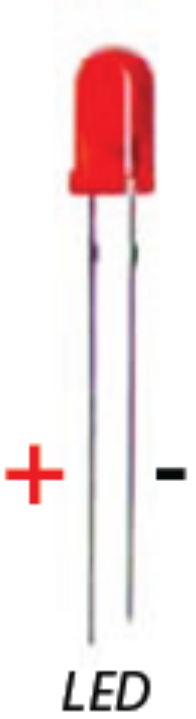
Outline

- **LED exercises**
- **WiFi-enabled LED matrix demo**
- **Use Javascripts to program LED matrix**
- **Use Arduino to program LED matrix**
- **Creative projects and presentation**

- Faculty and Volunteers
- Utility table

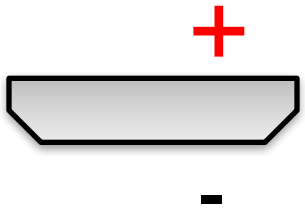
Basics of LED

- **LED stands for Light Emitting Diodes**
- Longer leg is **positive (+)**
- Shorter leg is **negative (-)**
  - Requires at least 1.8V to turn on, so a single AA/AAA battery (1.5V) cannot light it up...



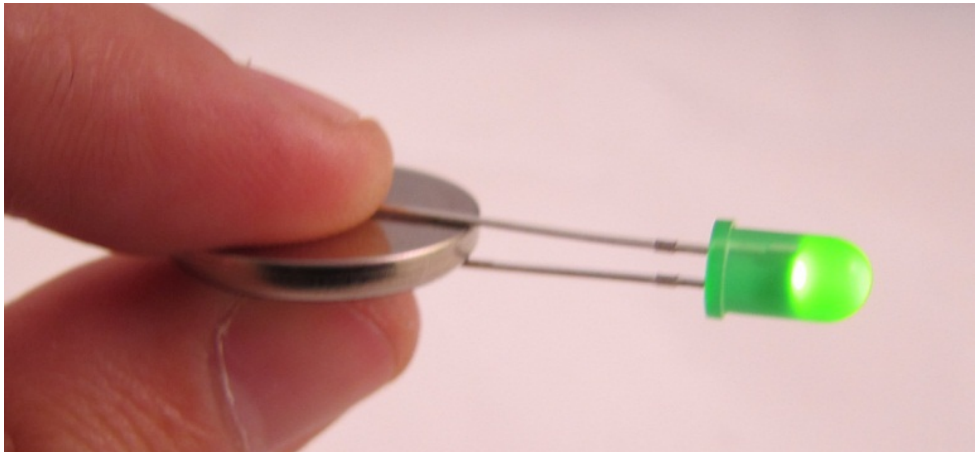
Basics of LED

- **Lithium Coin (button)** Battery
  - 3V typical voltage
  - Positive (flat) side is marked by **+**



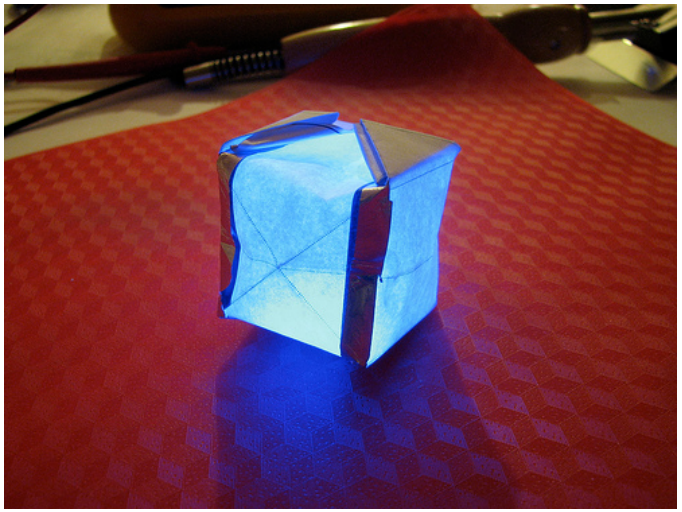
Basics of LED

- **Insert LED into coin battery (exercise)**
  - Take out a coin battery
  - ‘Sandwich’ the battery in between the two legs of LED: positive to +, negative to -
  - **Make sure not to short the legs!**

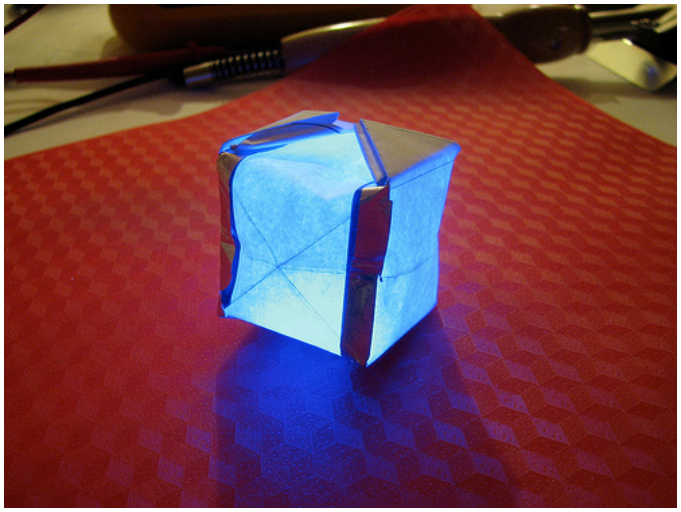


Basics of LED

- **Insert LED into coin battery (exercise)**
  - You can insert multiple LEDs!
  - Use tape to wrap the assembly and wear it!
  - More exciting types of LEDs
  - How long can this last?



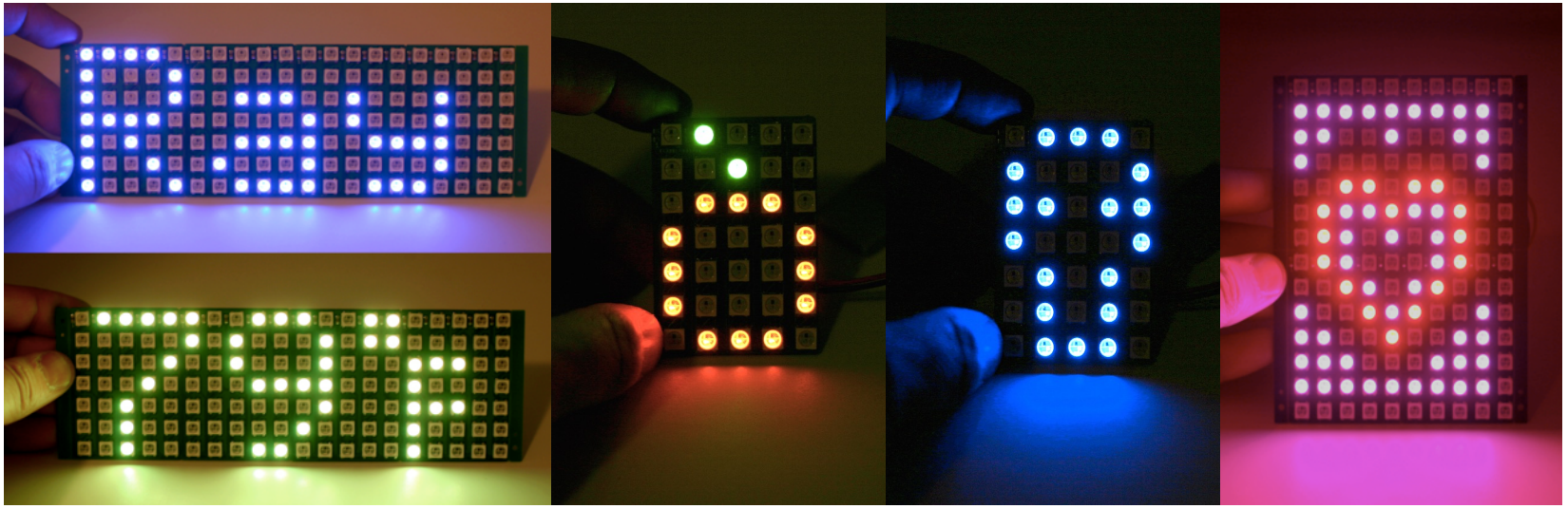
LED Accessories and Handicrafts



LED Blinkies

Basics of LED

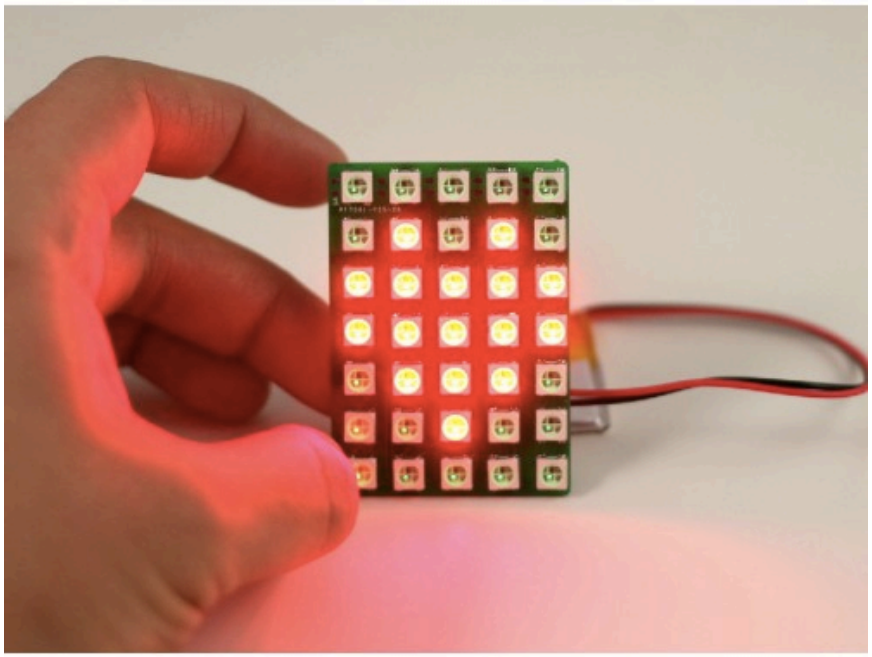
- A single LED is not that much fun
- Put a lot of LEDs together and arrange them as a matrix to make a display!



Color LED Matrix

- **35 Color LEDs arranged as a 7 (row) x 5 (col) matrix.**
- Column major order. First pixel marked on silkscreen.
- Neopixel: one input pin, and the color of every LED can be individually set.

0	7	14	21	28
1	8	15	22	29
2	9	16	23	30
3	10	17	24	31
4	11	18	25	32
5	12	19	26	33
6	13	20	27	34



Color 101

- Color is presented as R G B, each a brightness value between [0,255] (a byte).

```
Red:            (255,0,0)
Dark Red:       (64,0,0)
Green:          (0,255,0)
Blue:           (0,0,255)
White:          (255,255,255)
Gray:           (128,128,128)
Yellow:         (255,255,0)
what color?     (128,0,128)
```

Color 101

- Can also be represented as a long integer, in hexadecimal form (3 bytes):

```
Red:          0xFF0000
Dark Red:     0x400000
Green:        0x00FF00
Blue:         0x0000FF
White:        0xFFFFFF
Gray:         0x808080
Yellow:       0xFFFF00
Purple:       0x800080
```

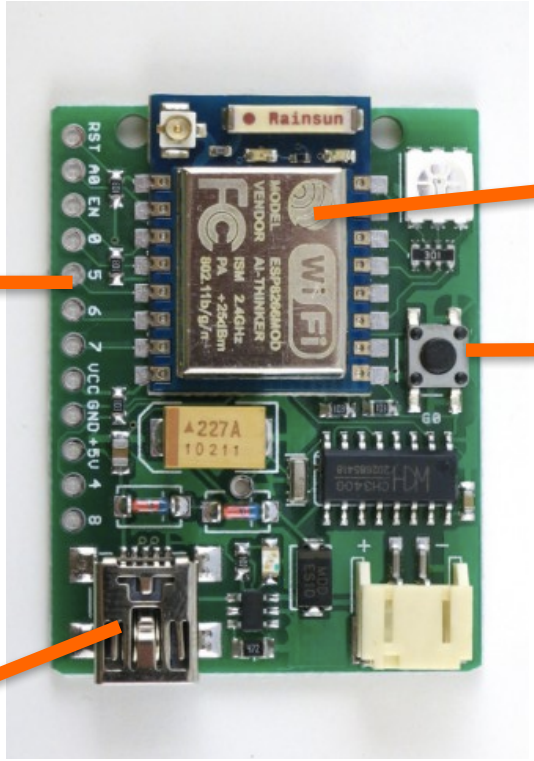
ESP8266 Microcontroller

- Extremely low cost ($2), tiny, with build-in WiFi.
- Arduino-compatible (using ESP8266 core library)
- Can be programmed wired through a USB port, or wirelessly through WiFi.

- 40MHz CPU
- 96KB main memory (RAM)
- 4MB flash (program) memory
- 802.11 b/g/n WiFi
- 9 digital GPIO + 1 analog pin
- Tons of WiFi-enabled projects built using it. Google ESP8266.

ESPtoy + LED Matrix

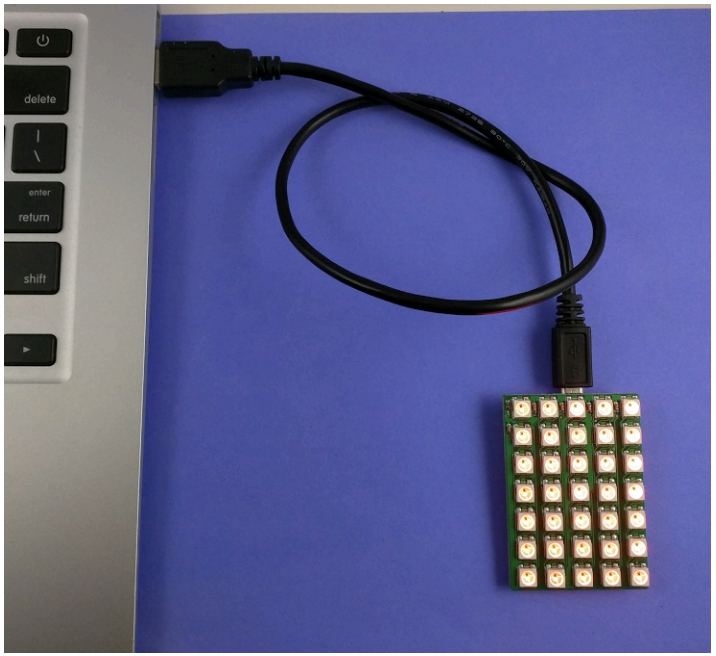
- At the back of the LED matrix is a stripped down version of ESPtoy – a prototype board for ESP8266



esptoy.rayshobby.net

Precautions

- Electronics are fragile. Please handle with care.
- Circuit has bare wires. Avoid using wet hands! **Keep work-area clean** of metal and conductive materials!
- Be gentle when plugging in the microUSB cable!
- LEDs can be bright, avoid staring at them for long.
- Cover with a white paper as a diffuser.
- Use Chrome or Firefox, no IE please.



Demo Time

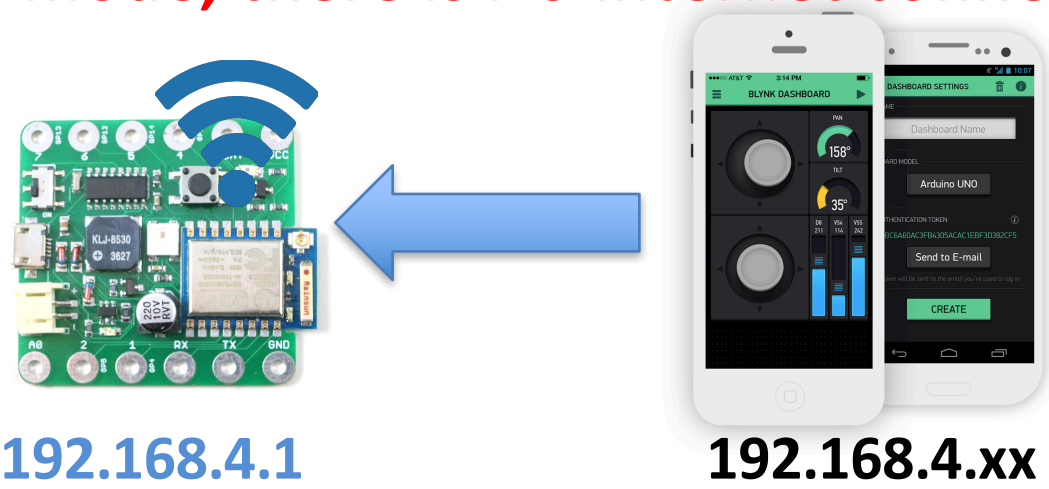
- Plug in microUSB cable
- Generally, board can be powered through computer's USB port, using a USB wall adapter or USB mobile charger / power bank.
- **Color Wipe**
- **Rainbow**

Demo Time

- **Webserver:**
  - Device creates a WiFi Access Point (AP): check the **back of the board** for your board's AP name.
    - **Please only connect to your own AP!**
    - Open WiFi (no password)
  - Use your computer (or phone) to connect to it, then open a browser and type http://192.168.4.1/
    - For Android phones, confirm 'Stay connected' warning by clicking on 'Yes'.
  - Explain pixel editing UI, Submit button, pre-defined patterns, show HTTP Get command.

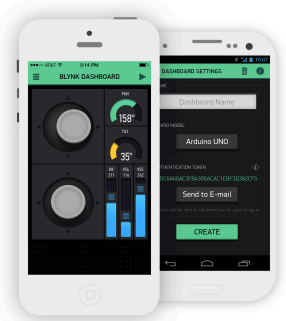
WiFi 101

- **AP (Access Point) mode**
  - Create a WiFi network with custom SSID & password
  - In this mode, ESP's default IP is 192.168.4.1
  - Other devices, such as your phone, can connect to this WiFi network, and access ESP using that IP.
  - **In this mode, there is No Internet connection!**

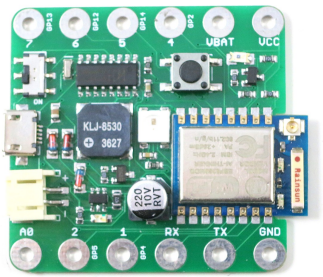


WiFi 101

- **Client mode**
  - Connects to an existing WiFi router and gets an IP assigned by your router. Can use your smartphone's WiFi hotpot too.
  - Has Internet connection.
  - Can upload data to cloud server or access remotely.



192.168.1.xx

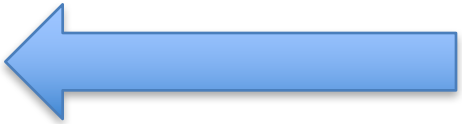


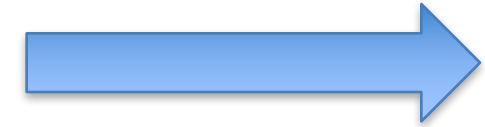
192.168.1.xx

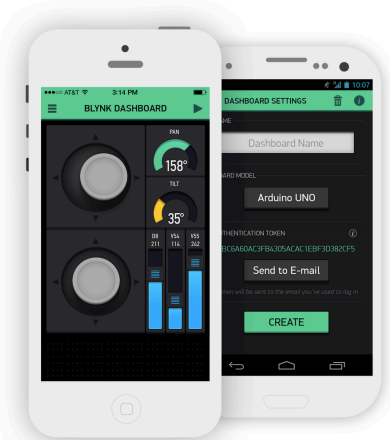
192.168.1.1

HTTP GET 101

http://192.168.4.1/







Firmware responds:

```
<html>
<title>Homepage</title>
…
</html>
```

HTML 101

- `1.HelloWorld.htm` (in Javascripts folder)

```
<html>
Hello World!
</html>
```

- The browser parses HTML and renders the result to the screen.
- Most browsers are very forgiving about the syntax and will do their best at interpreting the HTML.
- To edit code, use a decent text editor, such as Sublime Text, or Eclipse editor.

JavaScript 101

- 2.Write100.htm

```
<script>
for(i=0;i<100;i++)
   document.write('Hello  World!<br>');
</script>
```

- An interpreted languages that runs in a browser.
- Javascript is very flexible.
- Use Javascript to generate dynamic web content, or create a mobile app.

JavaScript 101

- `3.ConsoleDebug.htm`

```
<script>
for(i=0;i<100;i++)
   console.log(i);
</script>
```

- Most browsers support ‘developer tools’, where you can see error messages, print debugging information.

JavaScript 101

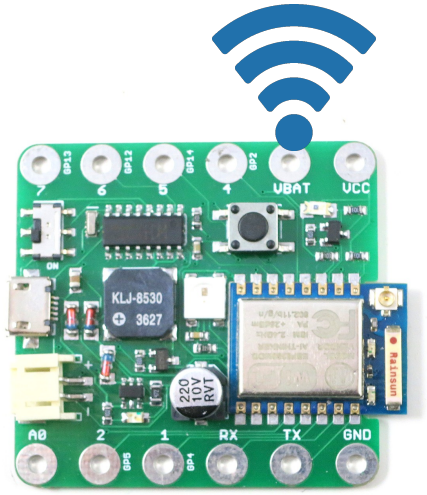
- `4.SimpleAnimation.htm`

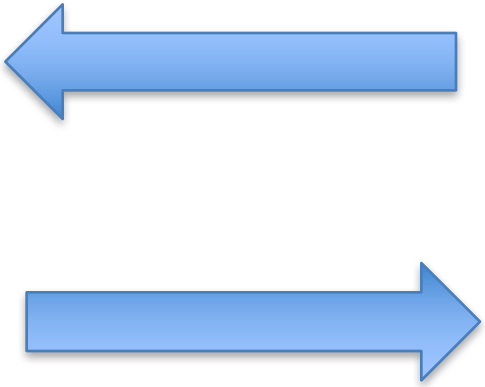
```
<script>
function animate() {
   …
}
setInterval(animate, 1000);
</script>
```

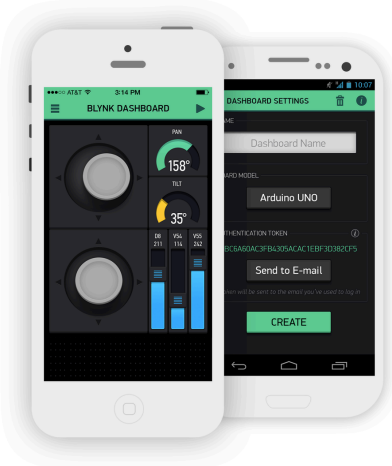
- Use `setInterval` method to repeatedly call a function that produces animation.

Get Server Variables / Status

http://192.168.4.1/js







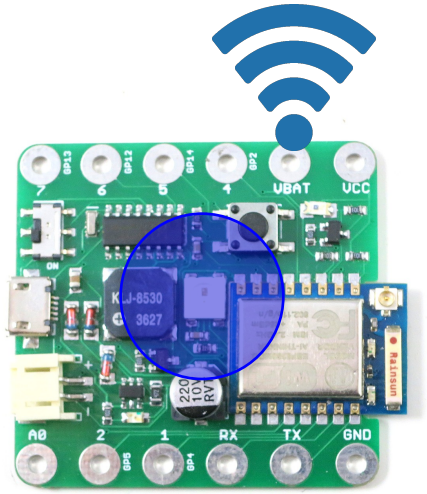
Firmware responds:

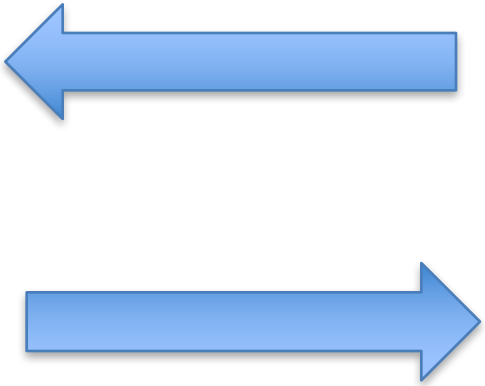
{“brightness”:48}

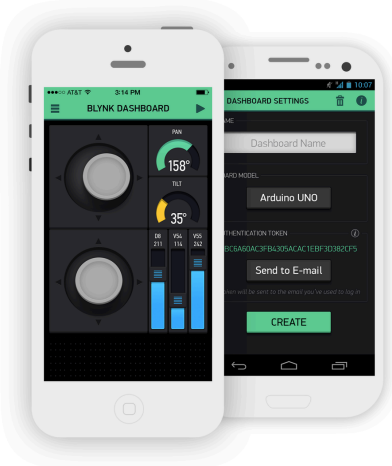
This is in JSON form

HTTP GET with Parameters

http://192.168.4.1/cc?brightness=20





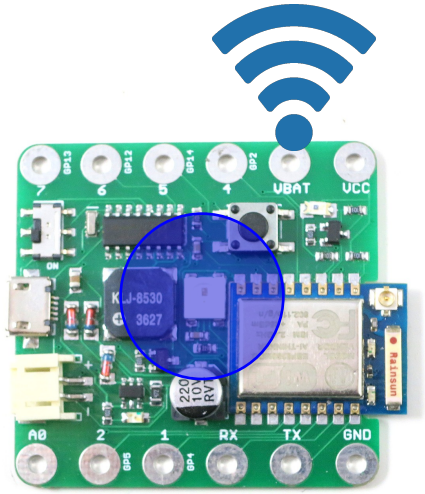


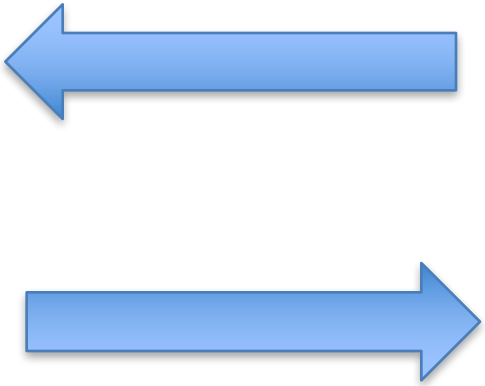
Firmware sets brightness and responds:

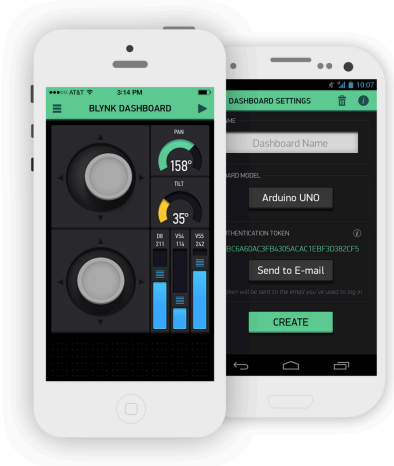
HTTP/200 OK
{“result”:1}

HTTP GET with Parameters

http://192.168.4.1/cc?pixels=00FF00…







Firmware sets pixels and responds:

HTTP/200 OK
{“result”:1}

Sets Pixels Colors

- 6 chars (hex color value) per pixel for each of the 35 LEDs (35x6=210 characters)
- Basically defines a complete 5x7 image

```
http://192.168.4.1/cc?pixels=000020000020
000020000020000020000020ff000000020ff000
000020ff0000ff0000ff0000ff0000ff0000ff00
00ff0000ff0000ff0000ff0000ff000000020ff0
000ff000000002000002000020ff000000002000
002000002000002000002000020000020
```

WiFiLED Demo API

- **Supported GET commands:**
- Return status:
  http://192.168.4.1/js
- Set brightness:
  http://192.168.4.1/cc?brightness=48
- Set pixel colors:
  http://192.168.4.1/cc?pixels=xxx
- Scroll text (with given color and speed):
  http://192.168.4.1/cc?scrolltext=CICS ROCKS!&color=002000&wait=75

Color and wait parameters are optional

Use JavaScript to Send Get Command

- 5.Heart.html

```
<script>
var xhr=new XMLHttpRequest();
xhr.open('GET','http://192.168.4.1/xxx',true);
xhr.send();
</script>
```

- JavaScript's way to send HTTP GET command.
- Many programming languages support this (e.g. Python), and even using Shell command!

Use JavaScript to Create Animations

- First, load a nice pixel pattern to your LED matrix (such as Heart), then try the following:
  `6.Blink.html`
- Close the browser tab after trying each demo to avoid it affecting the next demo.
  `7.Fade.html`
- Next, check out the following:
  `8.Animate.html`
  `9.Animate_gen.html`
  `a.Animate_pong.html`

Use JavaScript to Create Animations

- You can wrap the JavaScript you wrote in to a Mobile app (e.g. using PhoneGap), which you can load to your phone and run by simply clicking on it.

- Now we will give you some time to work on your own creative patterns or animations.

Programming WiFi LED Matrix

- Follow **ArduinoInstall.pdf**
- **Driver**
  - If using Mac or Windows 7/8, you need to install driver (see ArduinoInstall.pdf)
  - Linux does not need driver, but you need to run Arduino as ‘sudo’ or root.
  - ESP8266 supports Over-The-Air (OTA) programming (for advanced users).

Programming WiFi LED Matrix

- **Step 2: Bootloading Mode**
  - Unplug the USB cable (on the computer's end).
  - **Press and hold the push-button while plugging the cable back in,** then release the button.
  - The demo should stop running, meaning the device is in programming mode.
  - NOTE: you need to enter bootloading mode **every time you upload a new program** to the board.

Programming WiFi LED Matrix

- Launch Arduino, put board in **bootloading** mode
- **Tools -> Board -> Generic ESP8266 Module**
- **Tools -> Upload Speed -> 230400** (recommended)
- **Tools -> Port -> /dev/cu.wchusbserialxxx**
  or COM?
  or /dev/ttyUSB?
- **File -> Open**, select a demo program, say in Workshop folder -> Arduino -> blinkLED sub-folder.
- Click on **Upload**. This will take a while.

Entering Arduino-Land

- Setup() and Loop()

```
// include

// global variables

void setup()
{
   // initialization
}

void loop()
{
   // main loop
}
```

```
void main()
{
   setup();

   while(1) {
      loop();
   }
}
```

This is what happens internally

Serial Monitor

- Print and check data using **Serial Monitor**
- Very useful for debugging
- Set baud rate:
  ```
  Serial.begin(115200);
  ```

- Print data:
  ```
  Serial.println(“hello”);
  Serial.println(value);
  Serial.println(WiFi.localIP());
  ```

Useful Links

- ESP8266 core: github.com/esp8266/Arduino
  - ESP8266 specific language extensions
- ESP8266 community forum: esp8266.com
- Arduino: arduino.cc
  - Arduino programming language
- My hobby website: rayshobby.net
- Questions? Send to ruiwang@cs.umass.edu