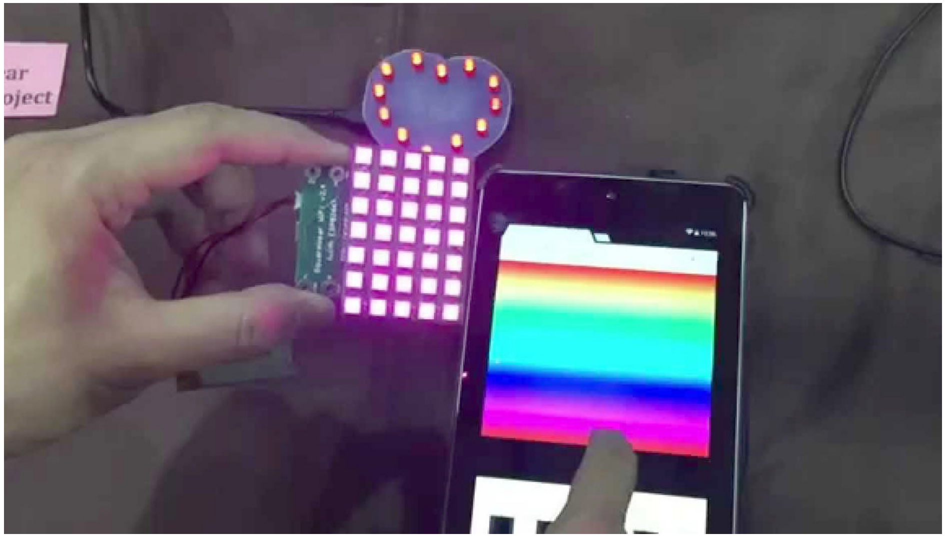
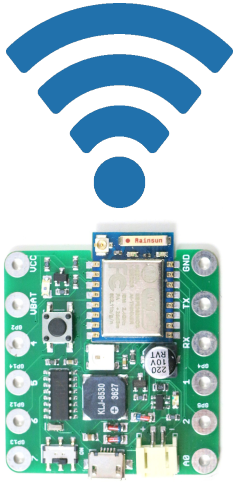
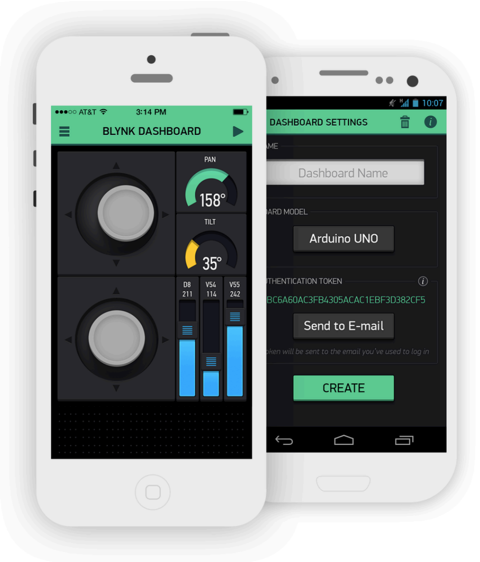






Making a WiFi Gadget using ESP8266

Rui (Ray) Wang

College of Computer and Information Sciences

UMass Amherst

Preparation

- Sit down and find a project partner at your table.
- Visit: graphics.cs.umass.edu/wearable/0416 download and unzip Arduino softwre, then follow INSTRUCTIONS.txt to install board profile. This will take a while! (Skip this step if you've already done it)
- Download and unzip workshop.zip from the above link. I will refer to this as the workshop folder (contains slides, sample code, driver).
- For Mac or Windows, install driver.
- If you don't have Internet connection, we have flash drives at each table.

Introduction

- **Goal**: learn the basics of making a WiFi gadget
  - Interact with it using a mobile phone or computer
  - Examples:
    - wearable LED badge
    - wireless sensors
    - interactive art exhibit
    - home automation, Internet of Things (IoT).
  - Combine programming with electronics to make something fun, physical, and useful.

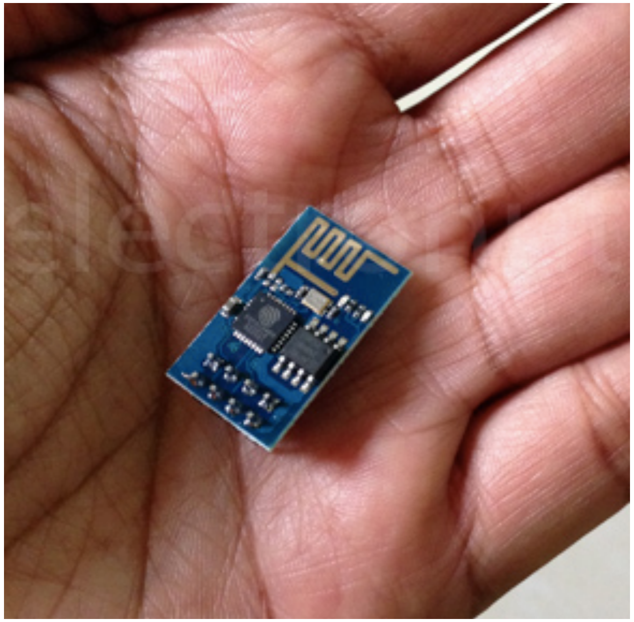
Introduction

- **Hardware**:
  - **ESP8266**: extremely low cost ($3), tiny, Arduino-compatible, 9 digital pins + 1 analog pin
  - Other options: Raspberry Pi, Arduino+WiFi shield, Arduino Yun…

- **Software**:
  - Arduino + ESP8266 core library

ESP8266

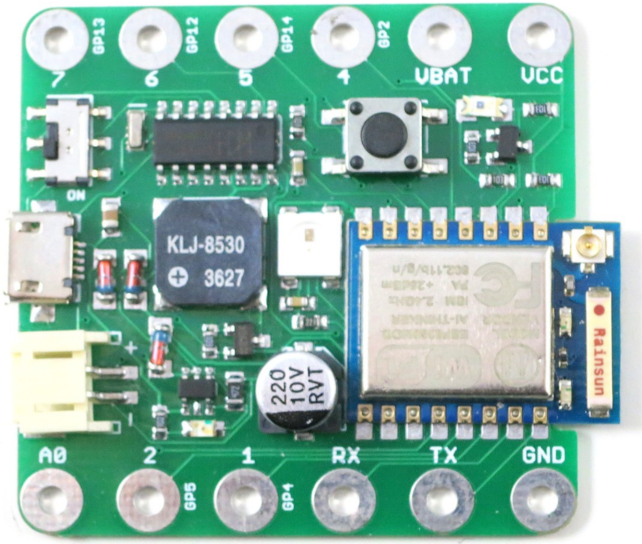
- 32-bit microcontroller
- 96KB main memory
- 512KB ~ 4MB flash (program) memory
- 802.11 b/g/n WiFi



- Tons of WiFi-enabled projects built using it. Google ESP8266 and you will find out.
- Community forum: *esp8266.com*

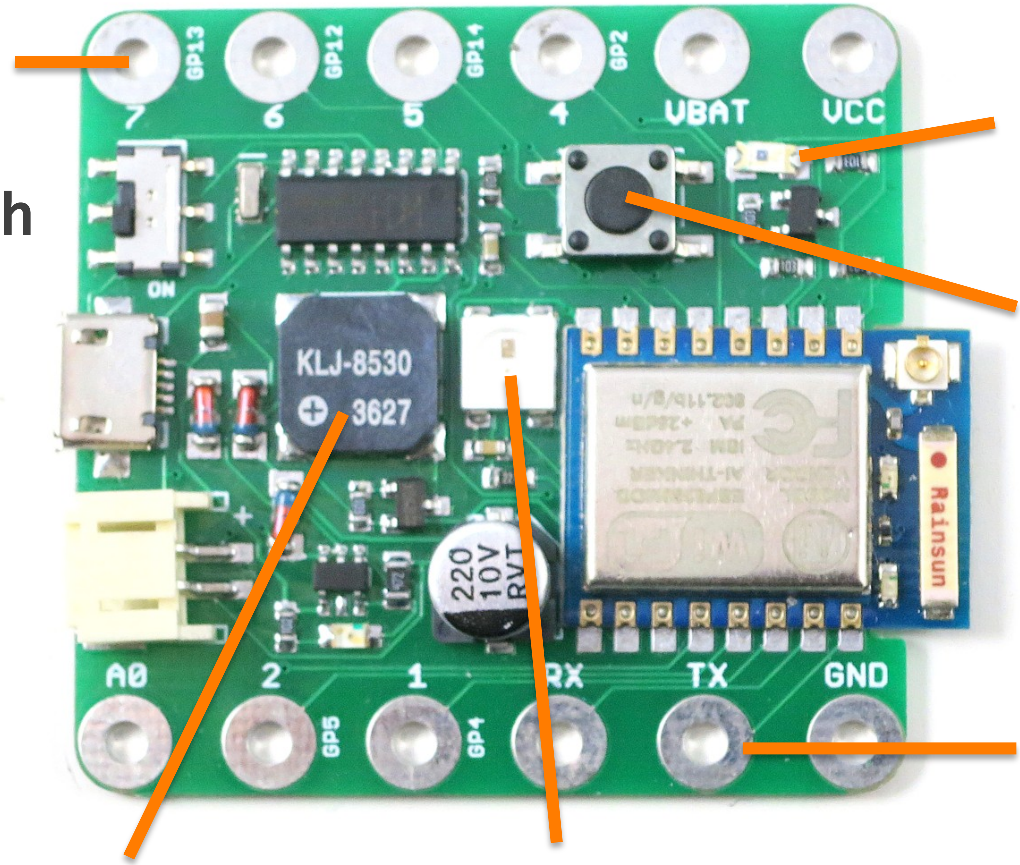
SquareWear WiFi

- ESP8266-based, with built-in components
  - Color LED, Button, Buzzer, Light sensor
  - microUSB port and USB-serial programmer
  - Large pin holes suitable for soldering, sewing, snaps
- Electronics are fragile, please handle with care: avoid shorting and handle gently!
- Website: sqrwear.com



SquareWear WiFi

Pins are also marked at the back of the board



Spare pins

Power switch

microUSB

Light sensor (enable: 16)

Button (0)

ESP8266

Spare pins

Buzzer (13)

Color LED (15)

SquareWear WiFi Demo

- Plug in microUSB cable and switch it on.
- **Buzzer**: start-up tune
- **Color LED**: blink, color wheel
- **Light Sensor**
- **Webserver:**
  - Device creates a WiFi Access Point (AP): check the back of the board for the AP name
  - Password is the same as the AP name (upper case!)
  - Use your phone / laptop to connect to it, then open a browser and type http://192.168.4.1/

Programming SquareWear WiFi

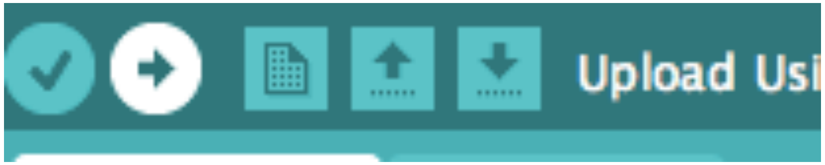
- **Step 1: Driver**
  - If using Mac or Windows (check the ‘drivers’ folder in workshop material).
  - Linux does not need driver, but you need to run Arduino as ‘sudo’ or root.
  - ESP8266 supports Over-The-Air (OTA) programming (for advanced users).

Programming SquareWear WiFi

- **Step 2: Bootloading Mode**
  - Switch off SquareWear WiFi.
  - **Press and hold the push-button while switching it back on,** 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 SquareWear WiFi

- 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 the blinkLED program, in Workshop folder -> 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

Useful Links

- Arduino: arduino.cc
  - Arduino programming language
- ESP8266 core: github.com/esp8266/Arduino
  - ESP8266 specific language extensions
- ESP8266 community forum: esp8266.com

Color 101

- Explain the **blinkLED** program
- Color is presented as R, G, B, each an intensity value between [0,255].

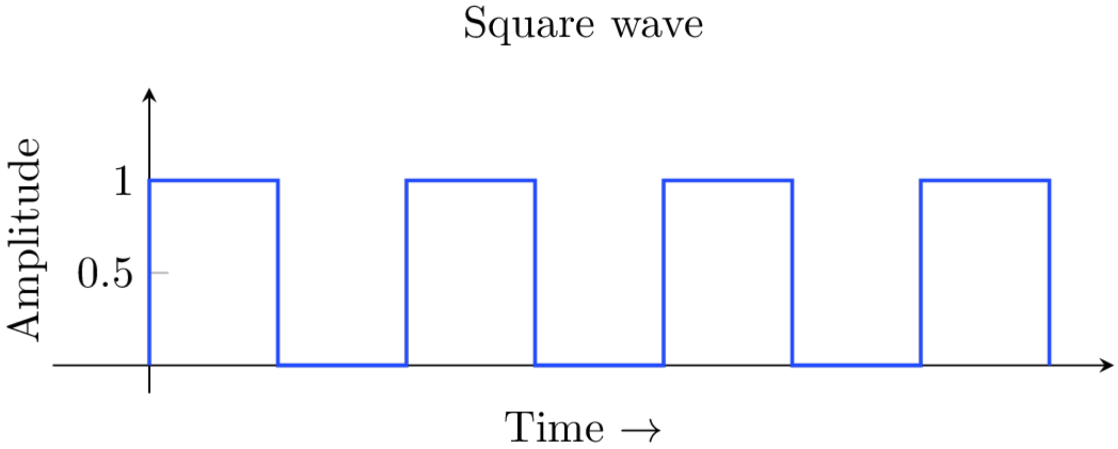
Red:	(255,0,0)
Dark Red:	(64,0,0)
Green:	(0,255,0)
Blue:	(0,0,255)
White:	(255,255,255)
Yellow:	(255,255,0)
?:	(128,0,128)

Color 101

- Neopixel (WS2812): chainable color LED
  - 1 digital pin to individually set the color of many LEDs.
- SquareWear has 1 neopixel on-board, wired to pin 15.
- You can also plug a color LED matrix at the back of the board (LED badge).

Sound 101

- What does a sound wave look like?



- Audible frequency is 20 Hz to 20K Hz
  - What's the frequency of A4 – the pitch standard?
- Explain the makeSound demo (warning!)

Digital Pin 101

- Most pins have multiple functions
- Set to input, and read a value (e.g. button)

```
pinMode(int pin, INPUT);
int digitalRead(int pin);
```

- Set to output, and write a value (e.g. relay)

```
pinMode(int pin, OUTPUT);
void digitalWrite(int pin, int value);
```

- There are several spare pins on SquareWear WiFi

Analog Sensor 101

- An analog pin that can read sensor values in **continuous scale**: from 0 to VCC (3.3~3.6V)

```
int analogRead(A0);
```

- **Light Sensor** (responds to ambient light)
  - Can use pin 16 to enable or disable it (allows sharing the analog pin with multiple sensors).
- Explain the lightSensor demo

Serial Monitor

- Print and check data using **Serial Monitor**
- Very useful for debugging
- Set baud rate:

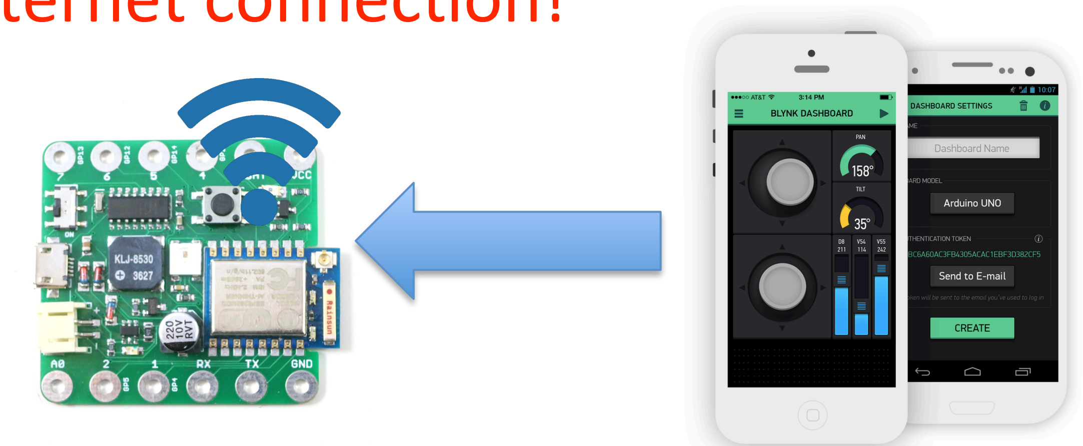
```
Serial.begin(115200);
```

- Print data:

```
Serial.println(“hello”);
Serial.println(value);
Serial.println(WiFi.localIP());
```

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 directly.
  - No Internet connection!

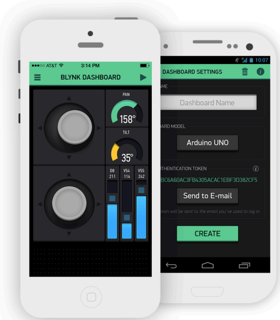


192.168.4.1

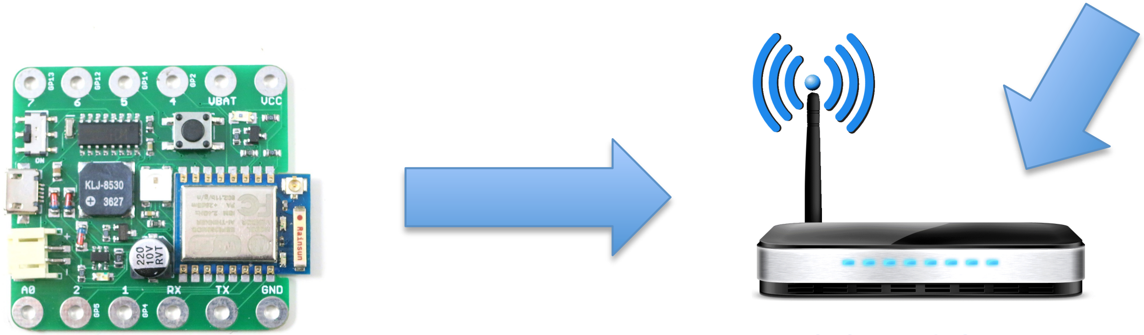
192.168.4.xx

WiFi 101

- **Client mode**
  - – Connects to an existing WiFi router and gets an IP assigned by your router.
  - – Has Internet connection
  - – Can upload data to cloud server
  - – Can access it remotely



192.168.1.xx



192.168.1.xx

192.168.1.1

HTML 101

- HelloWorld.htm

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

- The browser parses the 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.

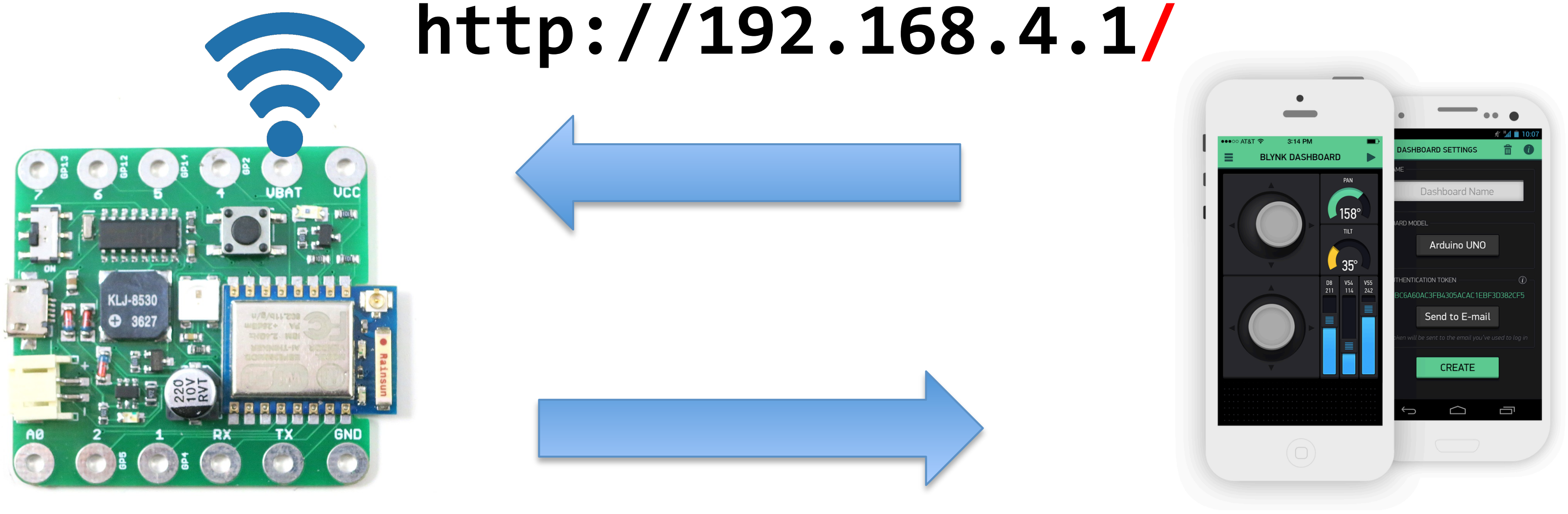
JavaScript 101

- Write100.htm

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

- Using Javascript you can generate dynamic webpage content .

HTTP GET -- Homepage

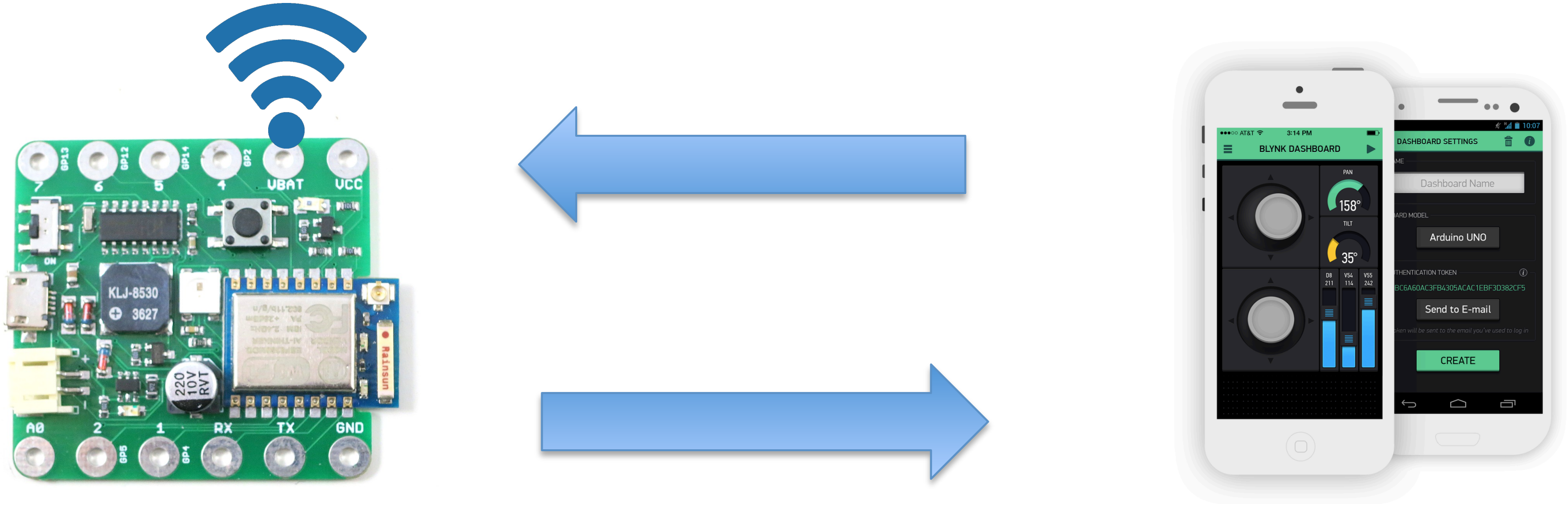


http://192.168.4.1/

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

HTTP GET – Custom Page

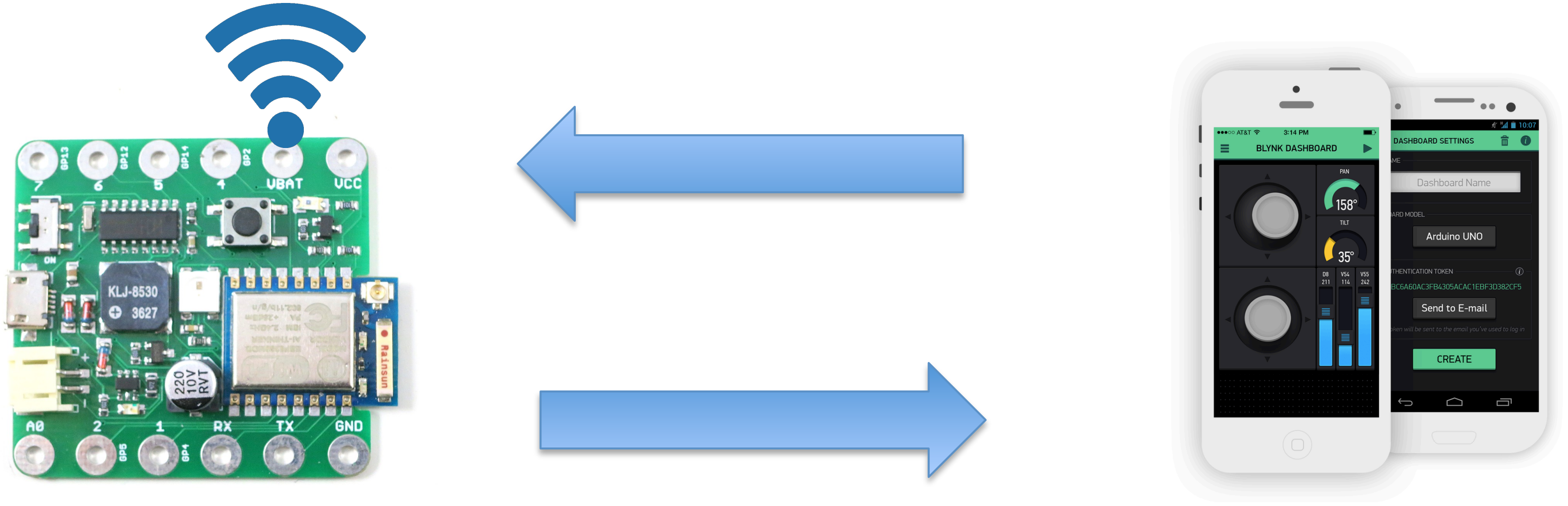
```
http://192.168.4.1/hello
```



```
<html>
<title>Hello Page</title>
…
</html>
```

Get Server Status / Variables

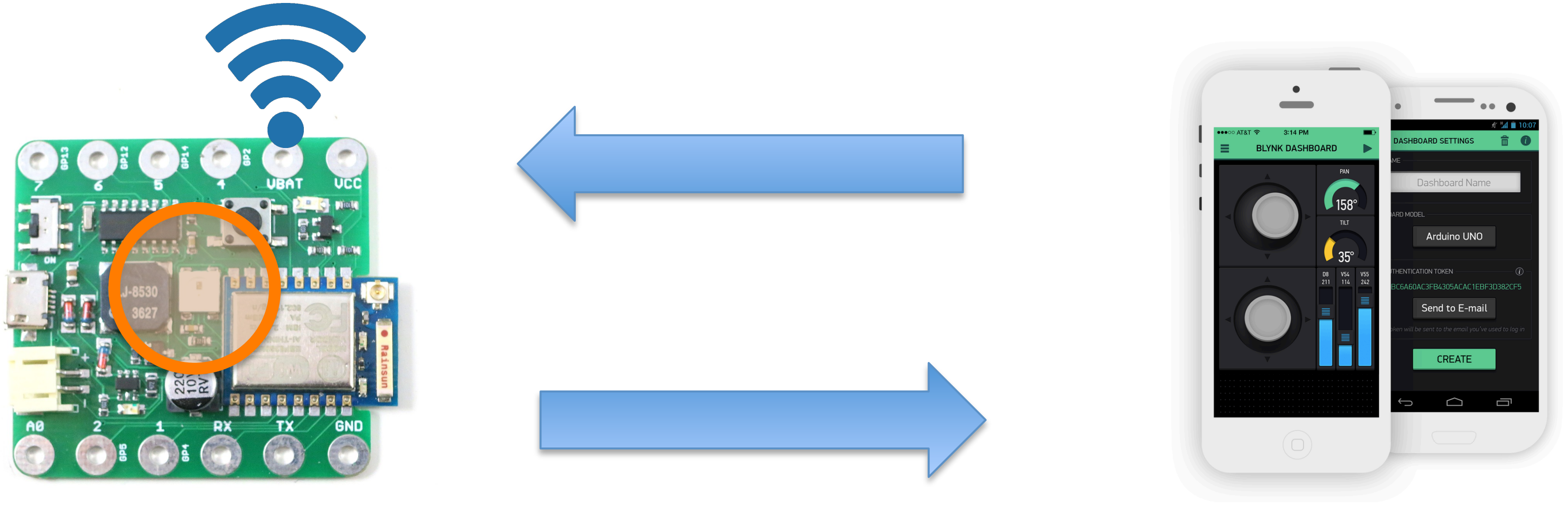
```
http://192.168.4.1/status
```



```
{“button”:1, “light”:450}
```

Trigger Action on HTTP GET

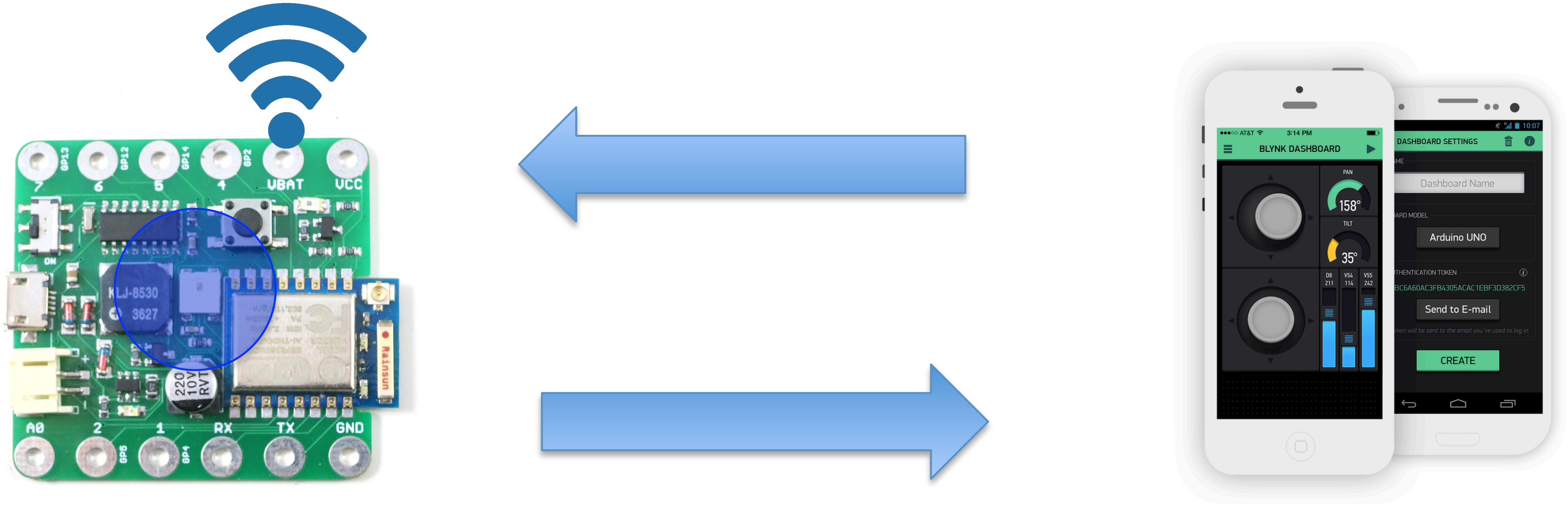
http://192.168.4.1/led



HTTP/200 OK

HTTP GET with Parameters

http://192.168.4.1/led?r=0&g=0&b=255



HTTP/200 OK

basicServer Demo

- **simpleServer demo (AP mode)**
  - The SSID is assigned based on the last 6 digits of the chip's MAC address.
  - Upload the demo
  - Use your phone (or computer) to connect to your custom WiFi (e.g. SW_xxxxxx)
  - Open a browser and try the following
    `192.168.4.1/`
    `192.168.4.1/js`
    `192.168.4.1/led`

basicAjax Demo

- You want the webpage to behave like an application: should look pretty and responsive!
- The HTML does the 'rendering' of the page, and dynamic content is retrieved asynchronously from the server through AJAX calls.
- This way, the server doesn't need to send the same HTML content over and over again, but instead only needs to send the relevant status variables.
- Explain the basicAjax demo

SWDemo

- Lastly, we turn back to the initial demo that came with the board: SWDemo

SquareWear WiFi Pins

- **Internally assigned pins:**
  - Color LED: digital 2 (Neopixel)
  - Button: digital 0 (bootloading button)
  - Buzzer: digital 15
  - Light sensor: digital 16 (enable)
    analog 0 (read value)
- **Available pins for general purpose use:**
  - 4, 5, 12,13, 14 (GP names)
  - Analog 0

Power Options

- SquareWear WiFi can be powered via
  - USB cable
  - USB mobile charger
- Interval voltage (VCC) is 3.3 ~ 3.5V.
- Digital HIGH -> 3.3V
  Digital LOW -> 0V