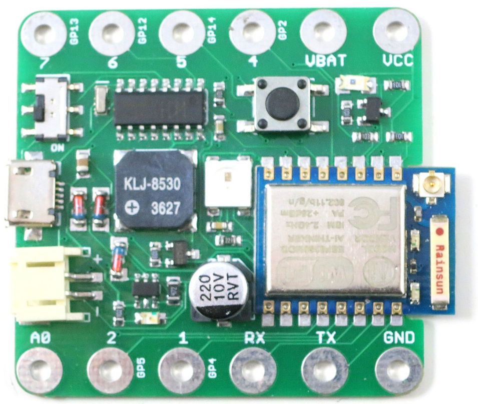
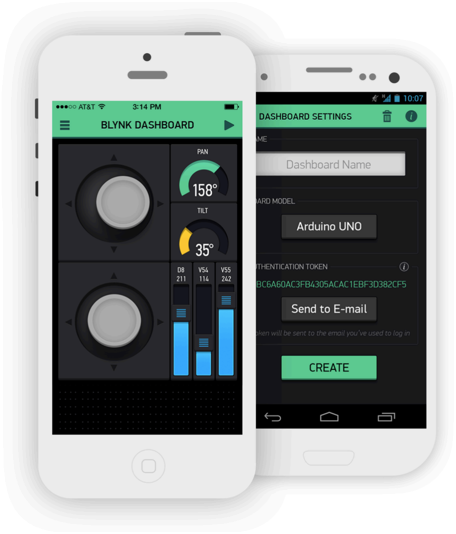




Arduino Programming using SquareWear WiFi

Rui (Ray) Wang

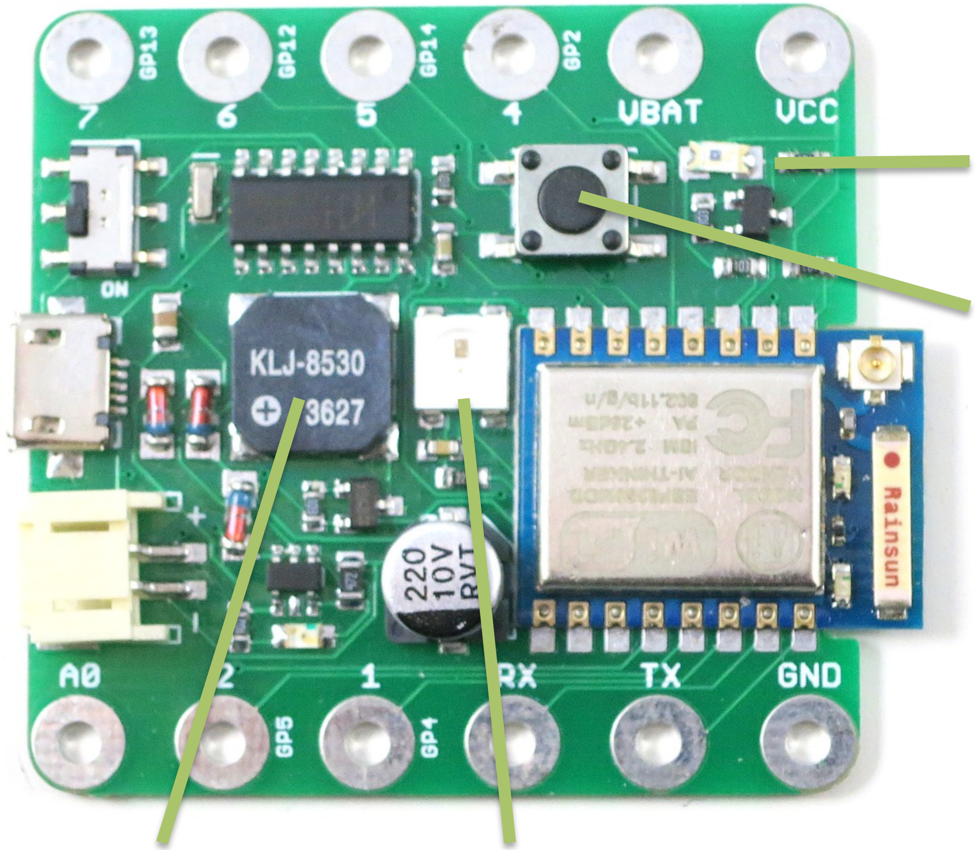
College of Computer and Information Sciences

UMass Amherst

Introduction

- **SquareWear WiFi** (http://sqrwear.com)
  - Wearable microcontroller with large pins
    - Suitable for sewing, soldering, touch sensing
  - WiFi capable, Arduino compatible
    - ESP8266 WiFi chip
  - Many built-in components
    - Color LED, Button, Buzzer, Light sensor
    - Battery jack, Charging circuitry
    - Built-in microUSB port for programming, Serial communication, and charging

SquareWear WiFi



Power switch

microUSB

Battery jack

Light sensor

Button (0)

WiFi chip

Buzzer (15)

Neopixel Color LED (2)

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
  - External Lithium battery
- Interval voltage (VCC) on the board is 3.3V.
- Digital HIGH -> 3.3V
  Digital LOW -> 0V

ESP8266

- A very low-cost WiFi-capable microcontroller.
  - 32-bit, 80MHz microcontroller (MCU)
  - 36KB RAM (memory)
  - 1~4MB flash (program mem)
  - WiFi 802.11 b/g/n/e/i
- Lots of community support
  - Many IoT gadgets built with ESP8266
  - Arduino support
  - Lots of firmwares (Lua, Python etc).

ESP8266

- Some differences with standard Arduino:
  - Voltage: 3.3V (vs. 5V on standard Arduino)
  - Digital pins: less, but more functionalities
  - Analog pins: only 1
  - Much faster, lots of RAM, flash, support for file system.

Arduino Programming Tips

- Setup() and Loop()

```
void setup()
{
   // initialization
}

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

```
void main()
{
   setup();
   while(1) {
      loop();
   }
}
```

This is what happens internally

ESP8266 Tips

- For long delays, **DO NOT** use delay (which will cause reset). Use **yield_delay(int ms)** instead.

`delay(1000);` → `yield_delay(1000);`

- Start-up Demo

Programming SquareWear WiFi

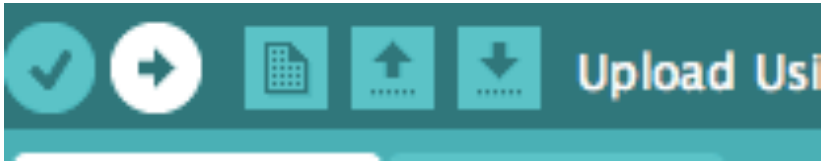
- **Enter Bootloading Mode**
  - Power off SquareWear by **sliding the power switch to OFF**
  - **Press and hold the push-button while powering it back on,** then release the button.
  - The demo should stop running, meaning the microcontroller is in programming mode.
  - You need to enter bootloading mode every time you upload a new program to the board.

Color LED

- Neopixel
- Blink
- Fade

Upload a Program

- A slightly customized version of Arduino 1.6.5 for SquareWear WiFi.
- **Tools -> Board -> Generic ESP8266 Module**
- **Tools -> Upload Speed -> 230400** (will make it faster to upload program)
- **Tools -> Port -> /dev/cu.wchusbserialxxx**
- **File -> Open**, select the blinkLED program, in Desktop -> SQWiFi -> blinkLED folder.
- Click on **Upload**.



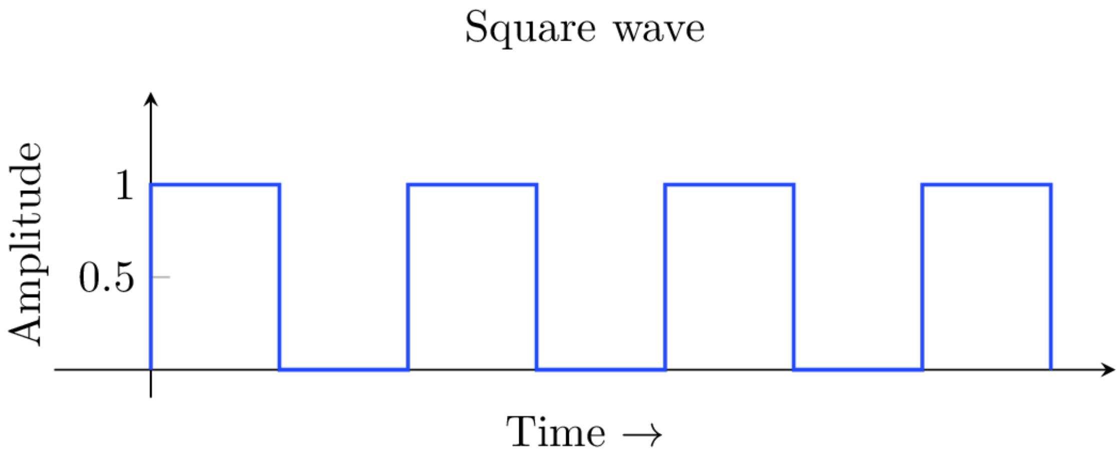
Color 101

- 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)

Make a Sound

- What does a sound wave look like?



- Audible frequency is 20 Hz to 20K Hz
  – What’s the frequency of A4 – the pitch standard?

Make a Sound

- Buzzer demo

Detect Button Click

- Button demo
- **pinMode(int pin, INPUT);**
- **int digitalRead(int pin);**
- Simple debounce implementation

Read a Sensor

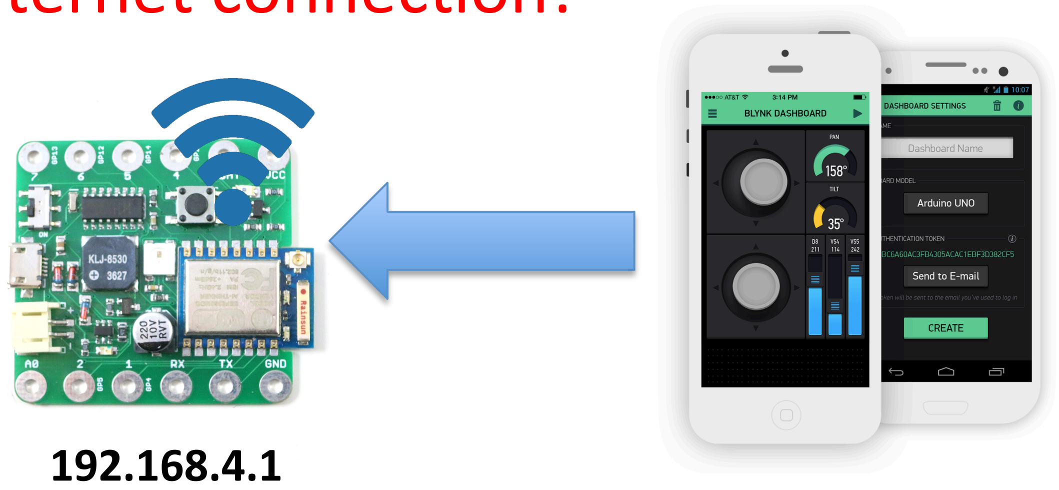
- An analog pin that can read sensor values in **continuous scale**: from 0 to VCC
- **int analogRead(int pin);**
- Only one analog pin marked **A0**
- **Light Sensor** (photo-transistor)
  - Responds to ambient light
  - Has an enable pin (to allow using the same analog pin for other sensors).

Serial Monitor

- Print debugging data using **Serial Monitor**
- Set baud rate:
  Serial.begin(115200);
- Print data:
  Serial.println(“hello”);
  Serial.println(value);
  Serial.println(WiFi.localIP());

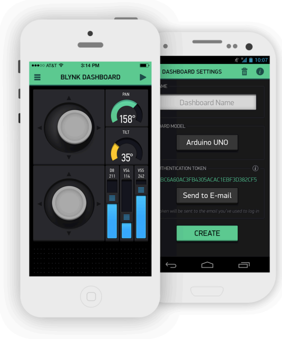
WiFi 101

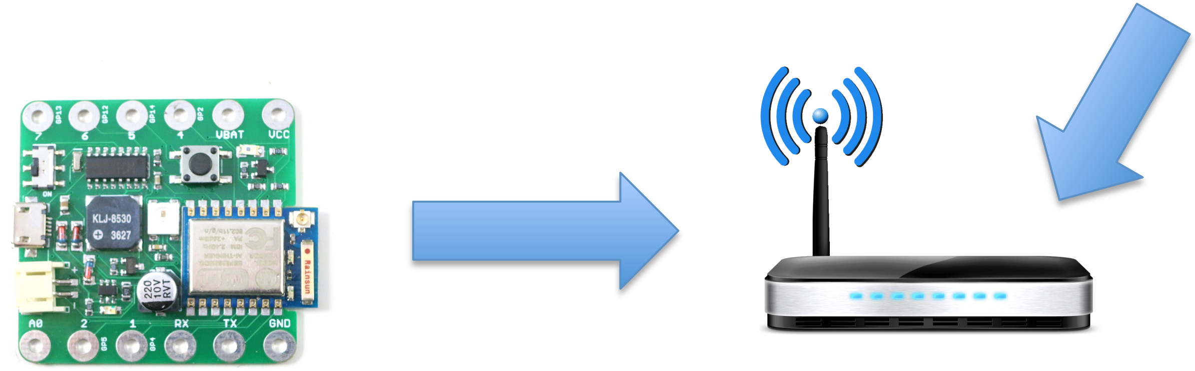
- **AP (Access Point) mode**
  – Create a WiFi network with custom SSID, password
  – In this mode, its default IP is 192.168.4.1
  – Other devices, such as your phone, can connect to this WiFi network, and access the AP directly
  – No Internet connection!



WiFi 101

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





192.168.1.xxx

WiFi 101

- **simpleServer demo (AP mode)**
  - **Change the SSID to a unique name**. A unique index is printed at the back of your board.
  - Upload the demo
  - Use your phone (or computer) to connect to your custom WiFi (e.g. SQWifi-xx)
  - Open a browser and type **192.168.4.1**
  - Then try **192.168.4.1/read**
  - And try **192.168.4.1/led**

WiFi 101

- Use AJAX, Javascript, and JSON to create dynamic web-based control
- **advancedServer demo (AP mode)**

WiFi 101

- Use Blynk App

Resources

- Arduino website
- ESP8266 for Arduino
- SquareWear