2017 MASTERs Conference

Getting Started with Johnny Five – The JavaScript Robotics and IoT Platform

MICROCHIP

Class Objectives

When you walk out of this class you will be able to…

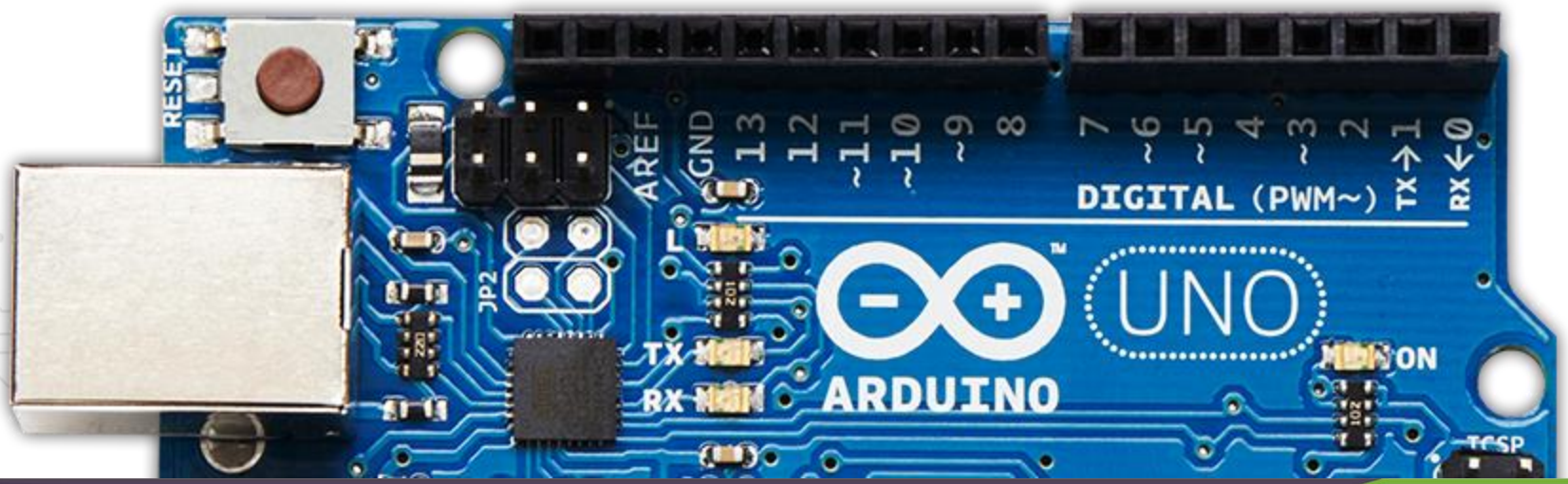
- **Program Arduino using Arduino Create and control using JohnnyFive**
- **Connect Arduino to the internet and control it remotely**
- **Publish sensor data to the web and visualize them using charts**

Agenda

- Intro to Johnny Five and Firmata
- Lab 1 – Hello World with Johnny Five
- IoT Concepts
- Lab 2 – Hello World for IoT
- Lab 3 – Sensor Data Visualization
- Summary

Arduino

- **Open-source prototyping platform based on easy-to-use hardware and software**
- **Write code and upload in minutes**



Firmata Protocol

- **Communicating with microcontrollers from software on a computer (or smartphone/tablet)**

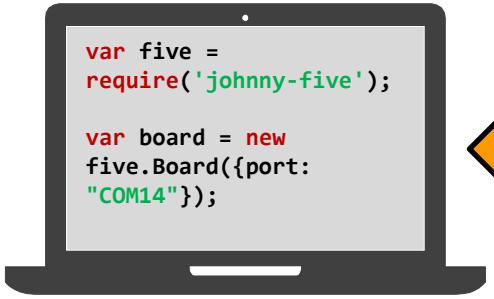
- **Methods of implementation**
  - Firmware of any microcontroller architecture
  - Any computer software package

Methods of Implementation of Firmata Protocol

- **Firmware of any microcontroller architecture**
  - E.g. StandardFirmata for Arduino and ChipKit platforms
- **Any computer software package**
  - Available in forms of multiple client libraries in different languages
  - Johnny Five is the JavaScript implementation of Firmata

Firmata Implementation

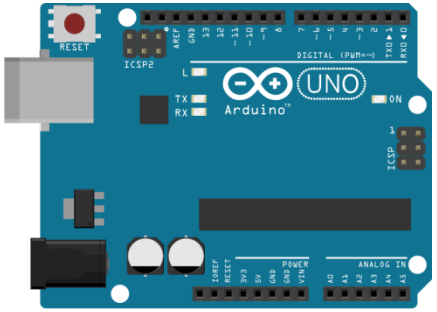
The Client runs a JavaScript program using Johnny Five



Client

Serial Communication

The Device runs a program using Firmata protocol



Device

What is Johnny Five

- Johnny Five is the JavaScript Robotics & IoT Platform
- Node.js based software framework
- Client Implementation of Firmata Protocol

Node.js

- **Platform for building fast and scalable network applications easily**
- **Uses an event-driven, non-blocking I/O model**
- **Lightweight and efficient**
  - Perfect for data-intensive real-time applications that run across distributed devices

But why JavaScript?!

Why not Embedded C?

- **Designed for IoT applications**
  - Applications easily connect to various web and cloud services using JavaScript
- **Allows software/web developers to program IoT devices without learning a new language**
  - Most web developers are already using JavaScript for their entire software stack

2017 MASTERs Conference

Lab 1

Hello World with Johnny Five

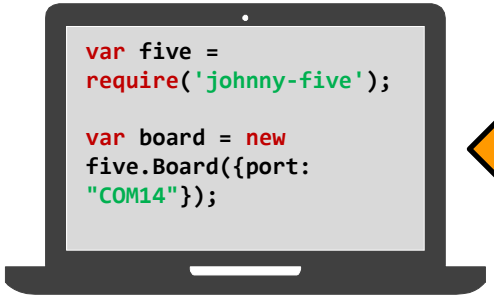
MICROCHIP

Lab 1 Objectives

- **Create a Hello World application using Johnny Five**
- **Blink an LED on Arduino Uno board while being controlled by Client**

Lab 1 Objectives

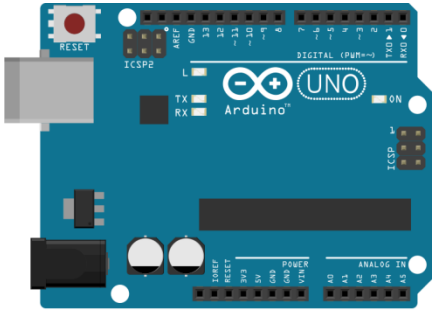
The Client runs a JavaScript program using Johnny Five



Client

Serial Communication

The Device runs a program using Firmata protocol



Device

Lab 1 Workflow

- System Setup
- Burn Firmata onto Arduino board
- Write client-side code using Johnny Five
- Execute program using Node

System Setup

- **Install Node.js**

  https://nodejs.org/

- **Install Johnny Five via NPM**

  ```
  npm install johnny-five
  ```

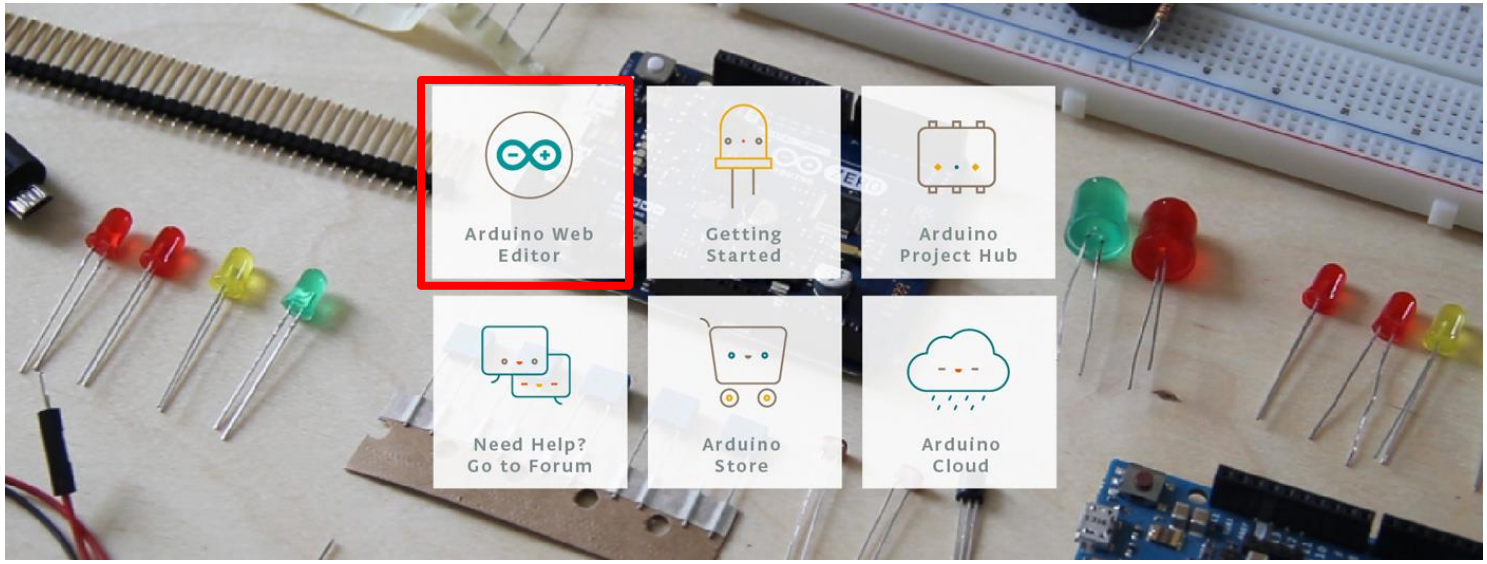
- **Install PubNub via NPM**

  ```
  npm install pubnub
  ```

Arduino Create

- **Integrated online platform to write code, access content, configure boards, and share projects**

https://create.arduino.cc/



Log into Arduino Web Editor

Username

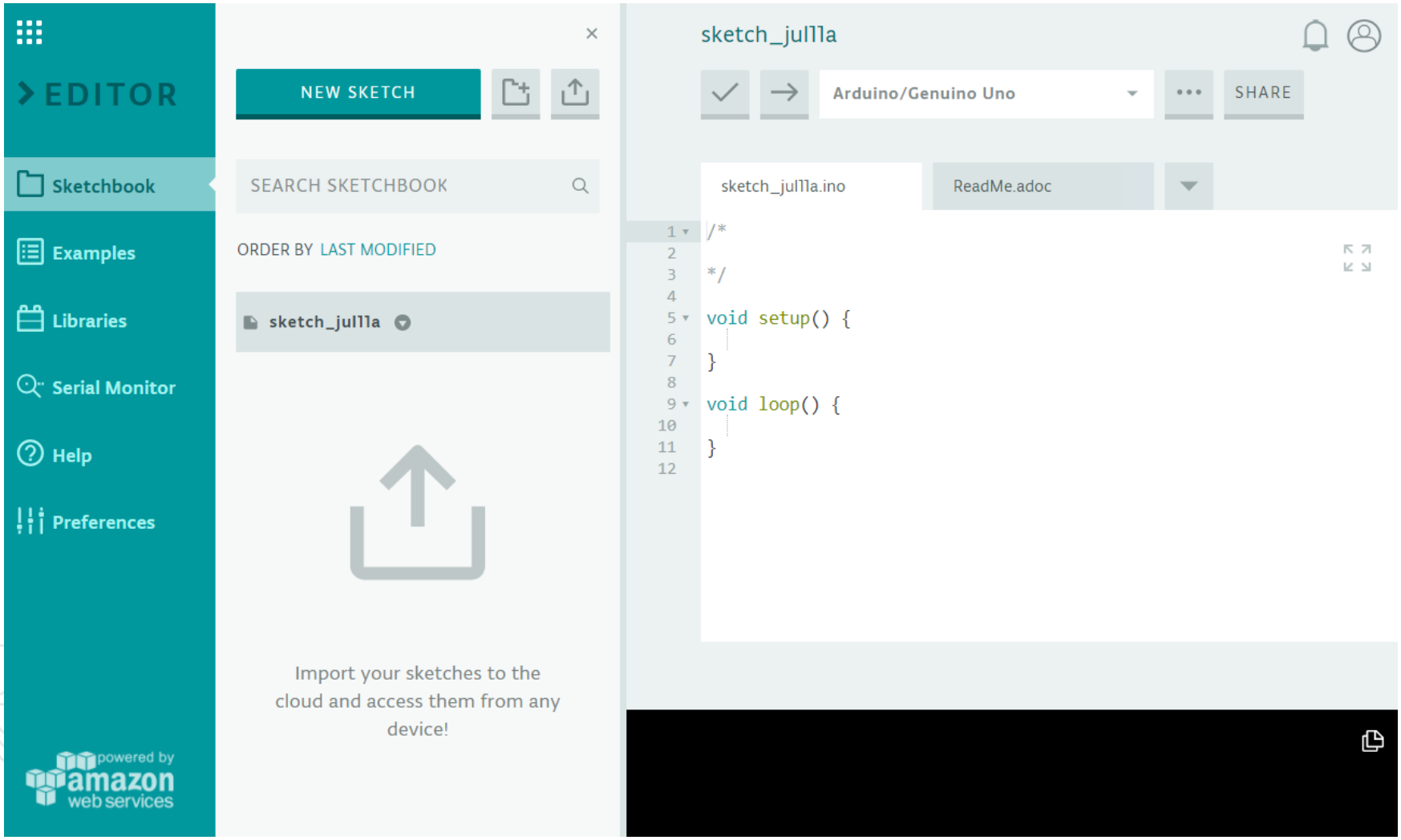
`microchiptestuser1`

Password

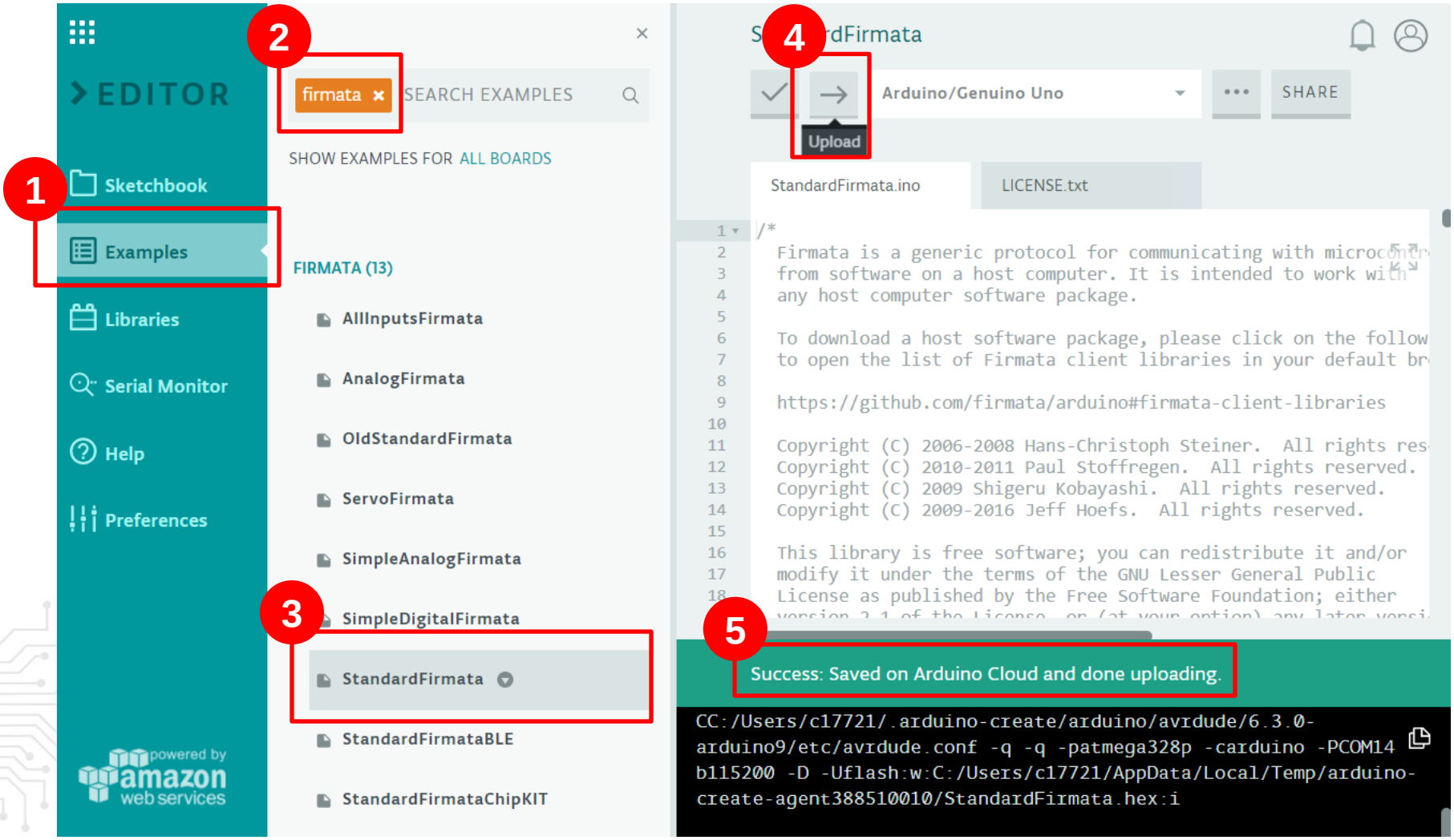
`masters2017`

Install Arduino Plugin

Arduino Web Editor



Upload Firmata onto Arduino



Blinking LED with Johnny Five

- **Create a file `blink.js` with this code:**

```
var five = require('johnny-five');
var board = new five.Board();

board.on('ready', function() {
  var led = new five.Led(13); // pin 13
  led.blink(500);             // 500ms interval
});
```

- **Run:**

```
$ sudo node blink.js
```

Lab 1 Summary

- Learned to use Arduino Web Editor
- Loaded Firmata onto Arduino
- Executed code using Johnny Five using Node

Pop Quiz 1

- **What is the JavaScript client implementation of Firmata called?**
  - (a) Node.js
  - (b) Johnny Five
  - (c) Arduino Uno
  - (d) Arduino Create

Pop Quiz 1

- **What is the JavaScript client implementation of Firmata called?**
  - (a) Node.js
  - *(b) Johnny Five* ☺
  - (c) Arduino Uno
  - (d) Arduino Create

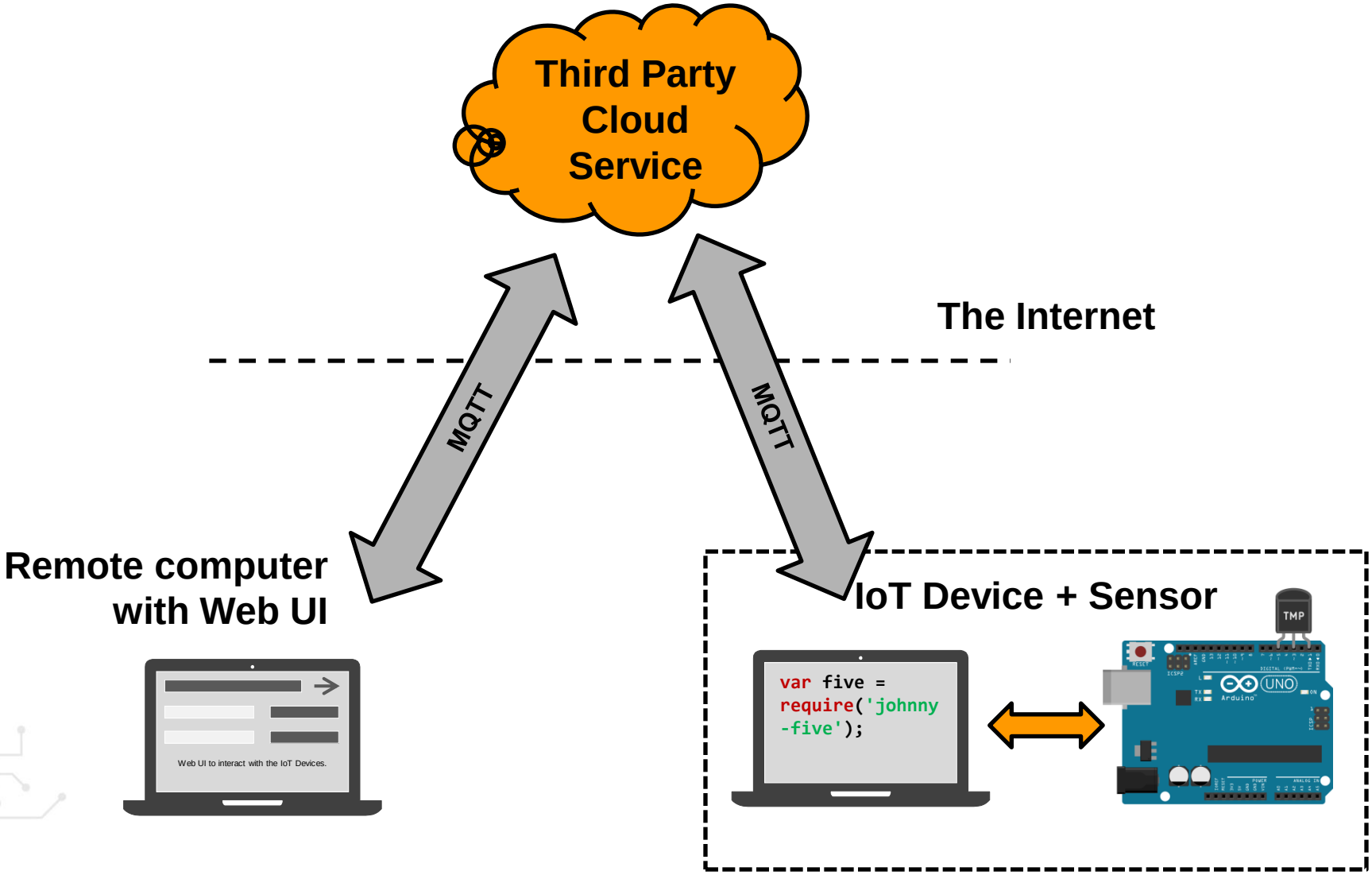
2017 MASTERs Conference

IoT Concepts

Connecting Arduino to the World using Johnny Five

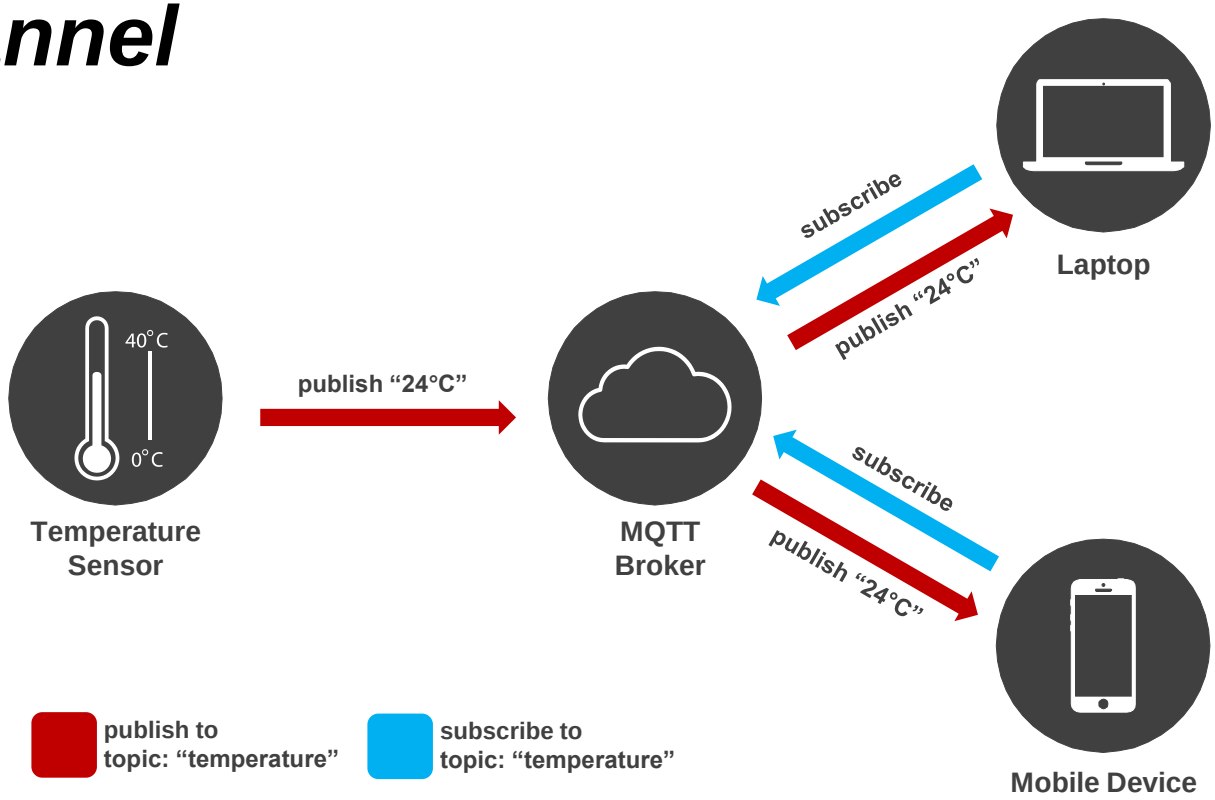
MICROCHIP

Basic IoT Setup



Pub/Sub Model

- ***Clients*** **decoupled via a** ***Broker***
- **Clients** ***publish*** **data or** ***subscribe*** **to a** ***topic/channel***



Browser-side Code

JS

HTML

CSS

Makes the items functional

Creates the structure of a web page

Makes the items look beautiful

2017 MASTERs Conference

Lab 2

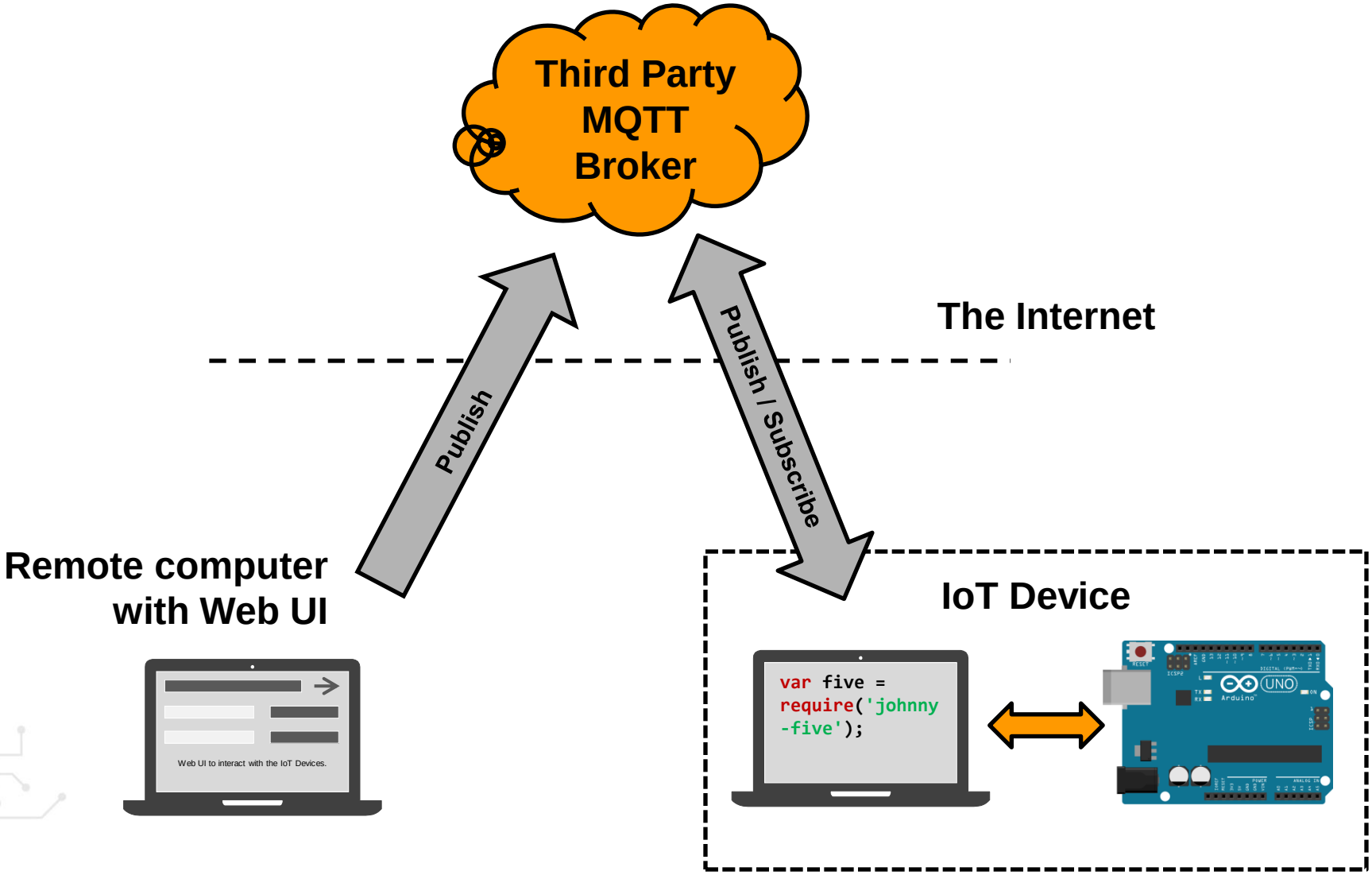
The Hello World of IoT

MICROCHIP

Lab 2 Objectives

- Create a Hello World application using IoT concepts
- Blink an LED on Arduino Uno board being controlled by a remote Client over the web

Lab 2 Objectives



MQTT Broker: PubNub

- **PubNub is a Data Stream Network for developers to build realtime apps**
- **Over 70 SDKs for mobile, web, desktop and server**
- **Get your free keysets from*:**

  https://www.pubnub.com/

*** For this class, use your assigned keysets from `pubnub-keysets.txt` file.**

Lab 2 // Part A

BROWSER-SIDE JAVASCRIPT

JavaScript SDK Basics

- **Install from CDN**

```
<script src =
"https://cdn.pubnub.com/sdk/javascript/pubnub.4
.8.0.js"></script>
```

- **Node.js**

```
$ npm install pubnub
```

Initialization

- **new creates an instance of the PubNub object for invoking PubNub operations**

```
var pubnub = new PubNub({
  subscribeKey: "sub-c-801ccfbc-...",
  publishKey: "pub-c-cfd2c879-...",
  ...
});
```

Publish

- **publish() is used to send messages to all subscribers of a channel**

```
pubnub.publish({
   channel: "temperature",
   message: {
         val: 22.5,
         unit: "celsius"
   }
});
```

JavaScript Object

Subscribe

- `subscribe()` causes the client to create an open TCP socket and begin listening for messages on a channel

```
pubnub.subscribe({
  channels: ["temperature", ... ],
  ...
});
```

Add Listener

- `addListener()` receives all the messages published on a channel

```
pubnub.addListener({
  message: function(m){ console.log(m) },
  error: function(e){ console.log(e) },
  ...
});
```

Browser-side Code

HTML

Creates the structure of a web page

JS

Makes the items functional

CSS

Makes the items look beautiful

Browser-side HTML

- Create a file **blink-remote-client.html** with:

```
<html><head>
  <title>Remote LED Blink</title>
  <script src="https://cdn.pubnub..."></script>
  <!-- Insert CSS styling here -->
</head>
<body>
  <h1>Prototyping IoT Demo UI</h1>
  <button class="on">Blink LED</button>
  <!– Insert Javascript code here -->
</body></html>
```

Browser-side CSS

- **Insert CSS in `blink-remote-client.html`:**

```
<!-- Insert CSS styling here -->

<style type='text/css'>
  button {
    font-size: 2em;
    padding: 10px 20px;
  }
</style>
```

Browser-side JS

- **Insert JS:**

```
<!-- Insert Javascript code here -->
<script type='text/javascript'>
  var pubnub = new PubNub({
      subscribeKey: "sub-c-801ccfbc-...",
      publishKey: "pub-c-cfd2c879-..." });
  var channel = "led";
  var button =
      document.querySelector("button.on");
  var blinkState = true;
  // Add button event listener here
</script>
```

Browser-side JS

- **Insert Listener:**

```
button.addEventListener("click", function(e){
    pubnub.publish({
        channel: channel,
        message: { blink: blinkState }},
    function(m) {
        console.log(m);
        blinkState = !blinkState;
        button.textContent = blinkState ?
                "Blink LED" : "Stop LED";
    });
});
```

Code Deployed

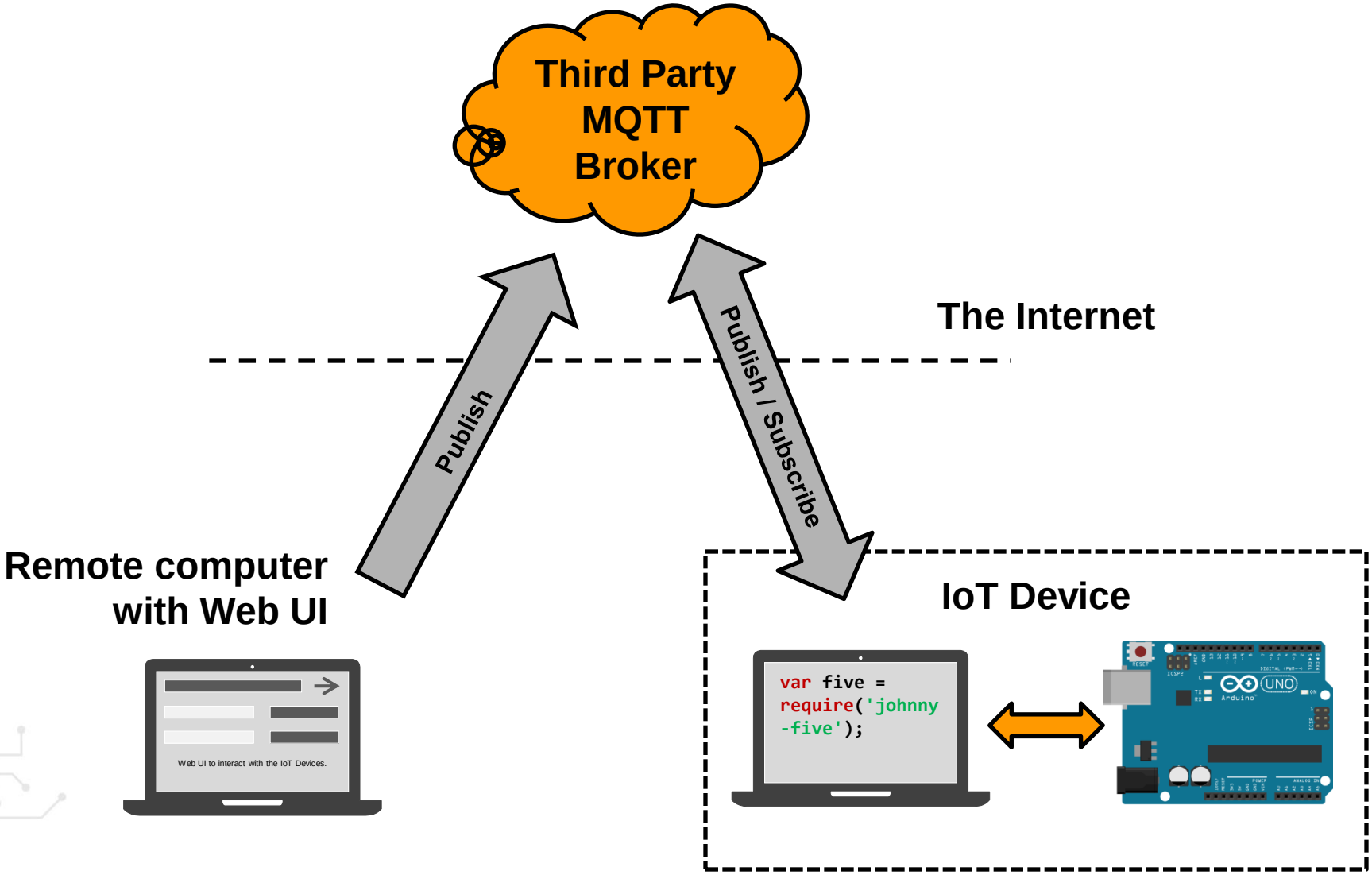
- **Code deployed at:**
  - https://codepen.io/maxmiaggi/pen/jmdyGX
  - https://maxmiaggi.github.io/blink-remote-client.html

MICROCHIP
MASTERs 2017

Lab 2 // Part B

SERVER-SIDE NODE.JS

Lab 2 Objectives



Server-side NodeJS

- **Add following in `blink-remote.js`:**

```
var channel = 'led';
board.on('ready', function() {
  var led = new five.Led(13); // pin 13
  pubnub.addListener({
    ...
  });
  pubnub.subscribe({
    channels: [channel],
  });
});
```

Server-side NodeJS

- **Add Listener in `blink-remote.js`:**

```
pubnub.addListener({
    message: function(m) {
      if(m.message.blink == true) {
        led.blink(500);
      } else {
        led.stop();
        led.off();
      }
    }
  });
```

Lab 2 Summary

- **Created a Hello World application using IoT concepts**
- **Controlled an LED on Arduino Uno board by a remote Client over the web**

Pop Quiz 2

- **Which of the following is not absolutely required for a web page to be functional?**
  - (a) HTML
  - (b) CSS
  - (c) JavaScript
  - (d) All of them are needed

Pop Quiz 2

- **Which of the following is not absolutely required for a web page to be functional?**
  - (a) HTML
  - *(b)* *CSS* ☺
  - (c) JavaScript
  - (d) All of them are needed

Pop Quiz 3

- **In the Pub/Sub Model, clients can subscribe to any number of channels at the same time.**
  - (a) True
  - (b) False

Pop Quiz 3

- **In the Pub/Sub Model, clients can subscribe to any number of channels at the same time.**
  - (a) *True* ☺
  - (b) False

2017 MASTERs Conference

Lab 3

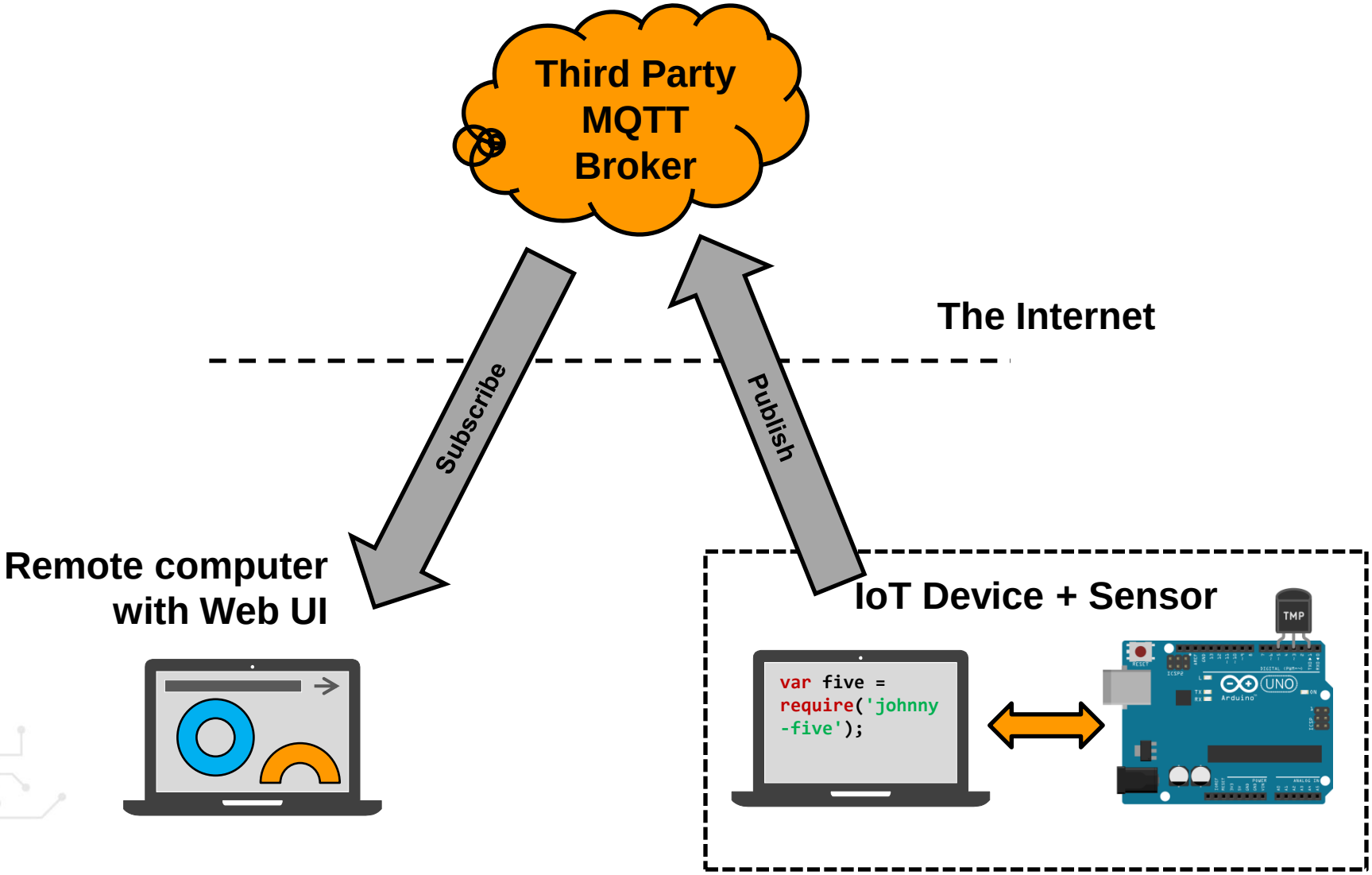
IoT: Sensor Data Visualizations

MICROCHIP

Lab 3 Objectives

- **Read sensor data from Arduino and publish to web**
- **Subscribe to the sensor data via a web page and create a visual using charts**

Lab 3 Objectives



What is EON

- Real-time visualization of data
- Open-source JavaScript Framework for Charting & Mapping
- Easy to get real-time data from PubNub channels
- Easy to use the data to plot graphs or maps

Plotting a Basic Chart

1. **Publish data to PubNub**
2. **Receive the data & display a chart**
   - Basic chart
   - Basic chart + history
   - Multiple graphs from multiple data
   - Customize the chart

Lab 3 // Part A

SERVER-SIDE NODE.JS

Publishing Data

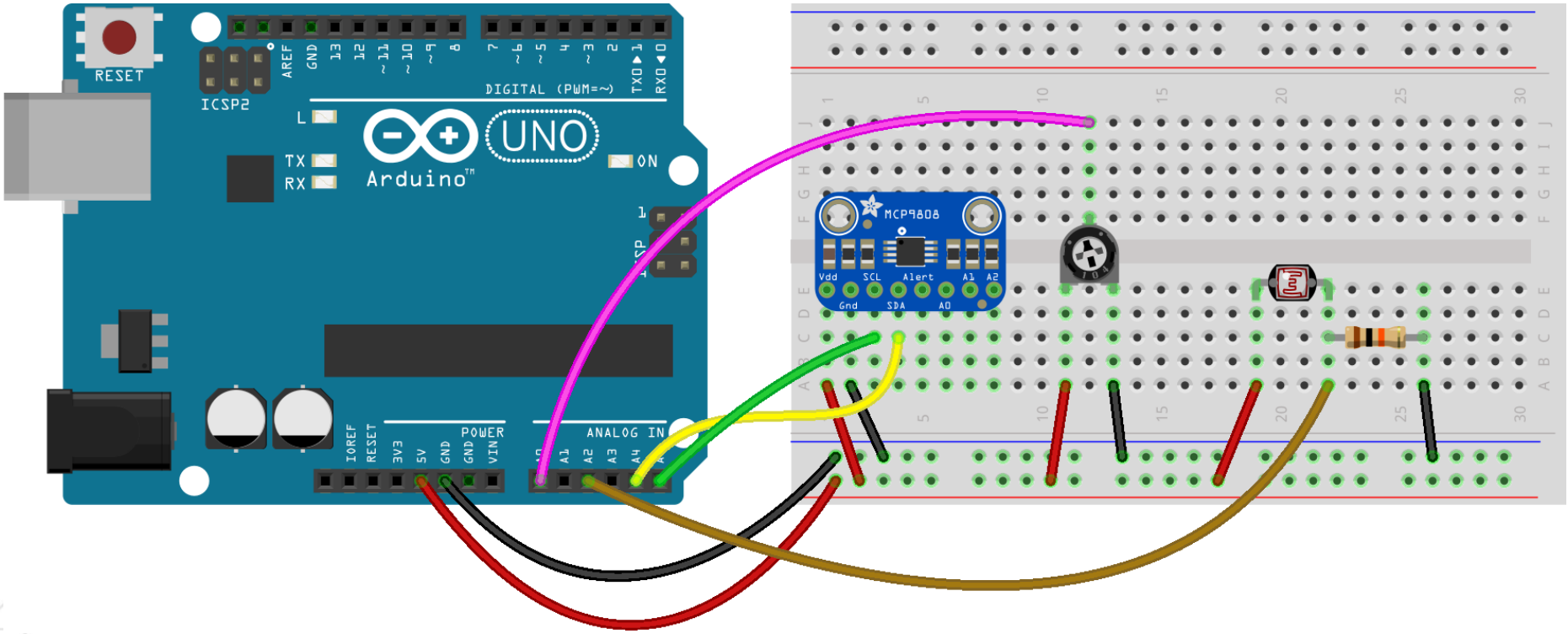
- **Initialize**

```
var PubNub = require('pubnub')();
var pubnub = new PubNub({
  subscribeKey: "sub-c-801ccfbc-...",
  publishKey: "pub-c-cfd2c879-...",
});
```

- **Publish some data**

```
pubnub.publish({
  channel: "my-data",
  message: { eon: {"humidity": 90} }
});
```

Circuit Diagram



MICROCHIP
MASTERs 2017

Lab 3 // Part B

BROWSER-SIDE JAVASCRIPT

Plotting a Line Graph

```
<html>
   <head>
         <script type="text/javascript"
src="http://pubnub.github.io/eon..."></script>
         <link type="text/css" rel="stylesheet"
href="http://pubnub.github.io/eon..." />
   </head>

   <body>
         <div id="chart"></div>
         <!-- Insert EON JavaScript here -->
   </body>
</html>
```

EON Browser-side JavaScript

```
<script type="text/javascript">
   var pubnub = new PubNub({
         subscribeKey : "sub-c-486be1d0-..."
   });

   eon.chart({
         channels    : ["random1"],
         generate    : { bindto: '#chart' },
         pubnub      : pubnub
   });

</script>
```

Using PubNub History API

```
eon.chart({
        channels    : ["random1"],
        generate    : { bindto: '#chart' },
        history     : true,
        pubnub      : pubnub
});
```

Publishing Multiple (Fake) Data

```
var data = { eon: {
        'value0' : Math.floor(Math.random()*99),
        'value1' : Math.floor(Math.random()*99),
        'value2' : Math.floor(Math.random()*99),
        'value3' : Math.floor(Math.random()*99),
        'value4' : Math.floor(Math.random()*99)
   }
}
pubnub.publish({
   channels  : ['random5'],
   message   : data,
});
```

Plotting Multiple Graphs

```
var pubnub = new PubNub({
        subscribeKey : "sub-c-486be1d0-..."
  });

  eon.chart({
        channels    : ["random5"],
        generate    : { bindto: '#chart' },
        pubnub      : pubnub
  });
```

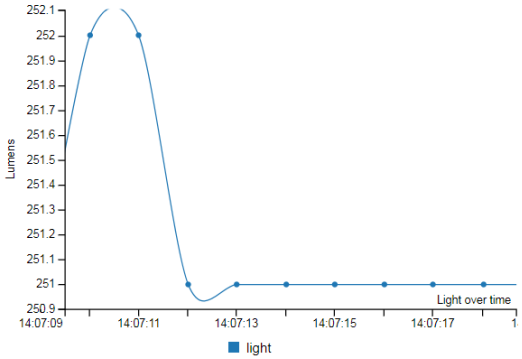
Customizing & Styling Data

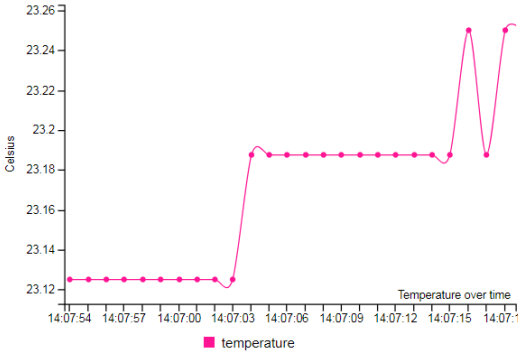
```
eon.chart({
   channels    : ["random1"],
   generate    : {
         bindto: '#chart',
         axis: { x: {
               type: 'timeseries',
               tick: { format: '%H:%m:%S' } },
         y: {
               label: { text: 'Temperature',
                        position: 'outer-middle'}
         } },
   }, ...
```

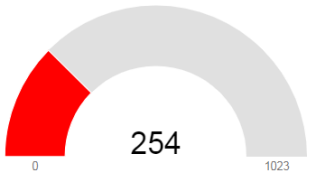
Code Deployed

- **Code deployed at:**
  - https://codepen.io/maxmiaggi/pen/bRZKYv
  - https://maxmiaggi.github.io/lab3-visuals.html

Sensor Data from Arduino







Lab 3 Summary

- **Read sensor data from Arduino and publish to web**

- **Subscribed to the sensor data via a web page and created a visual using charts**

Pop Quiz 4

- **In order to stream data in real-time and plot charts, what should a client be doing?**
  - (a) Publish to the channel(s)
  - (b) Subscribe to the channel(s)
  - (c) Both, publish and subscribe to the channel(s)
  - (d) Do nothing

Pop Quiz 4

- **In order to stream data in real-time and plot charts, what should a client be doing?**
  - (a) Publish to the channel(s)
  - *(b) Subscribe to the channel(s)* ☺
  - (c) Both, publish and subscribe to the channel(s)
  - (d) Do nothing

Summary

LEGAL NOTICE

SOFTWARE:

You may use Microchip software exclusively with Microchip products. Further, use of Microchip software is subject to the copyright notices, disclaimers, and any license terms accompanying such software, whether set forth at the install of each program or posted in a header or text file.

Notwithstanding the above, certain components of software offered by Microchip and 3rd parties may be covered by "open source" software licenses – which include licenses that require that the distributor make the software available in source code format. To the extent required by such open source software licenses, the terms of such license will govern.

NOTICE & DISCLAIMER:

These materials and accompanying information (including, for example, any software, and references to 3rd party companies and 3rd party websites) are for informational purposes only and provided "AS IS." Microchip assumes no responsibility for statements made by 3rd party companies, or materials or information that such 3rd parties may provide.

MICROCHIP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY DIRECT OR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND RELATED TO THESE MATERIALS OR ACCOMPANYING INFORMATION PROVIDED TO YOU BY MICROCHIP OR OTHER THIRD PARTIES, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBLITY OF SUCH DAMAGES OR THE DAMAGES ARE FORESEEABLE. PLEASE BE AWARE THAT IMPLEMENTATION OF INTELLECTUAL PROPERTY PRESENTED HERE MAY REQUIRE A LICENSE FROM THIRD PARTIES.

TRADEMARKS:

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.