

The background of the slide features a large, faint, circular seal of Brigham Young University (BYU). The seal contains the text "BRIGHAM YOUNG UNIVERSITY" around the top, "1888" in the center, "BYU" in large letters below the center, "IDAHO" below that, and "PROVO, UTAH" at the bottom. The seal is partially obscured by a diagonal blue band that runs from the top right towards the bottom left.

ECEN 361

Real-Time and Embedded Systems

Lab-10 Setup Description

ESP8266 Board and Cloud IOT

BRIGHAM YOUNG
UNIVERSITY

IDAHO



Lab-10 ESP8266

ESP8266 (Made by ExpressIF and ...)

- Cheap, reliable
- Wireless :WiFi, B/T
- 160Mhz Processor
- GPIOs, serial port USB
- Very popular
- Used by hundreds of mfg's (mostly Chinese)



Lab Description

Two Parts to the lab:

1. **No Wifi** – Just get a “BLINK” example running
2. **Wifi.** – connected IOT Device to Cloud Service (using Blynk.cloud)

Needed:

- ESP8266 board (NodeMCU or equiv)
- Cell phone
- WiFi network for which you have password and can join

Note: **BYUI** and **BYUI_Visitor** on campus are blocked

Lab-10 Part 1 -- Just running 1/3

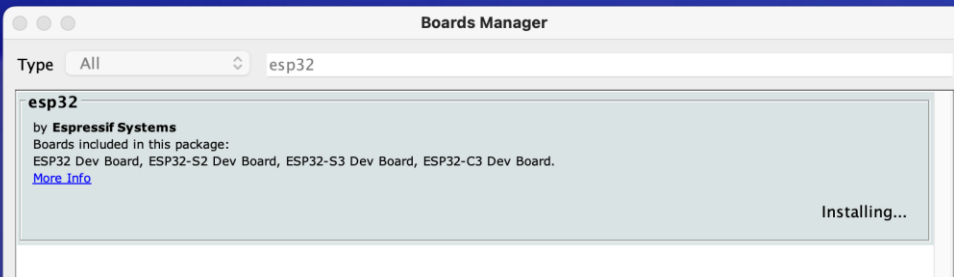
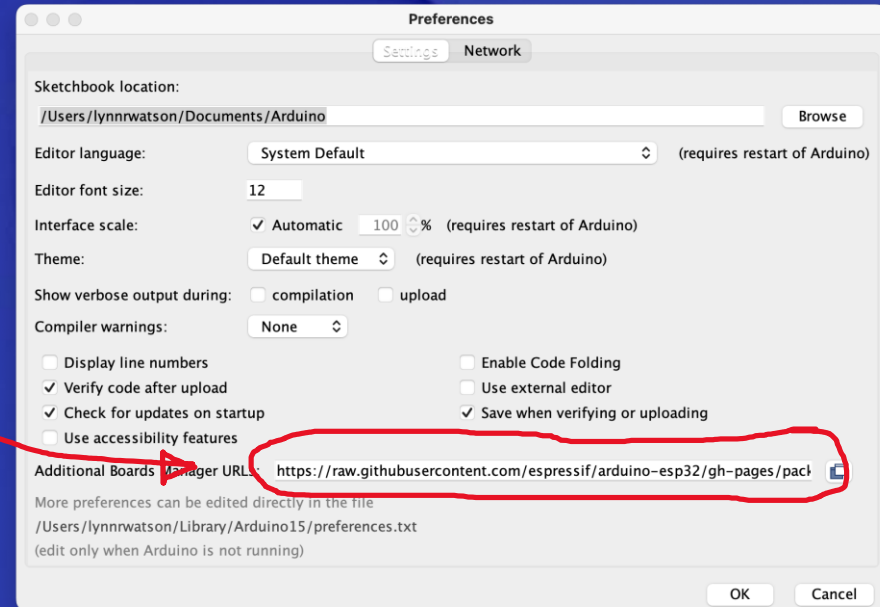
Steps: (Tutorial Online [HERE](#))

1. Arduino on your PC/Mac

2. Add the
Board Support Package:

1. Add [Library Path](#)

2. Add Board



Lab-10 Part 1 2/3

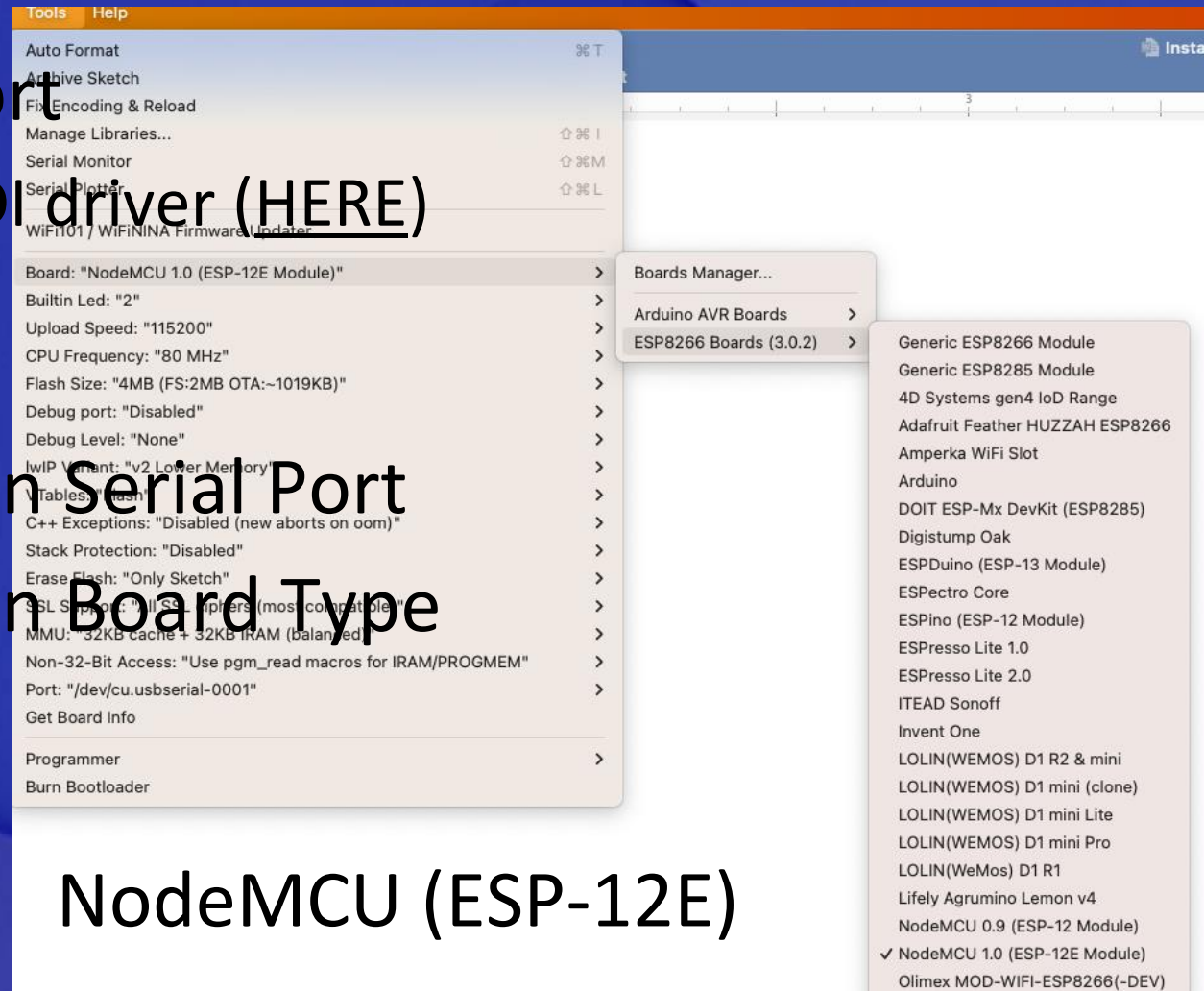
3. Check serial port

If not – Install FTDI driver ([HERE](#))

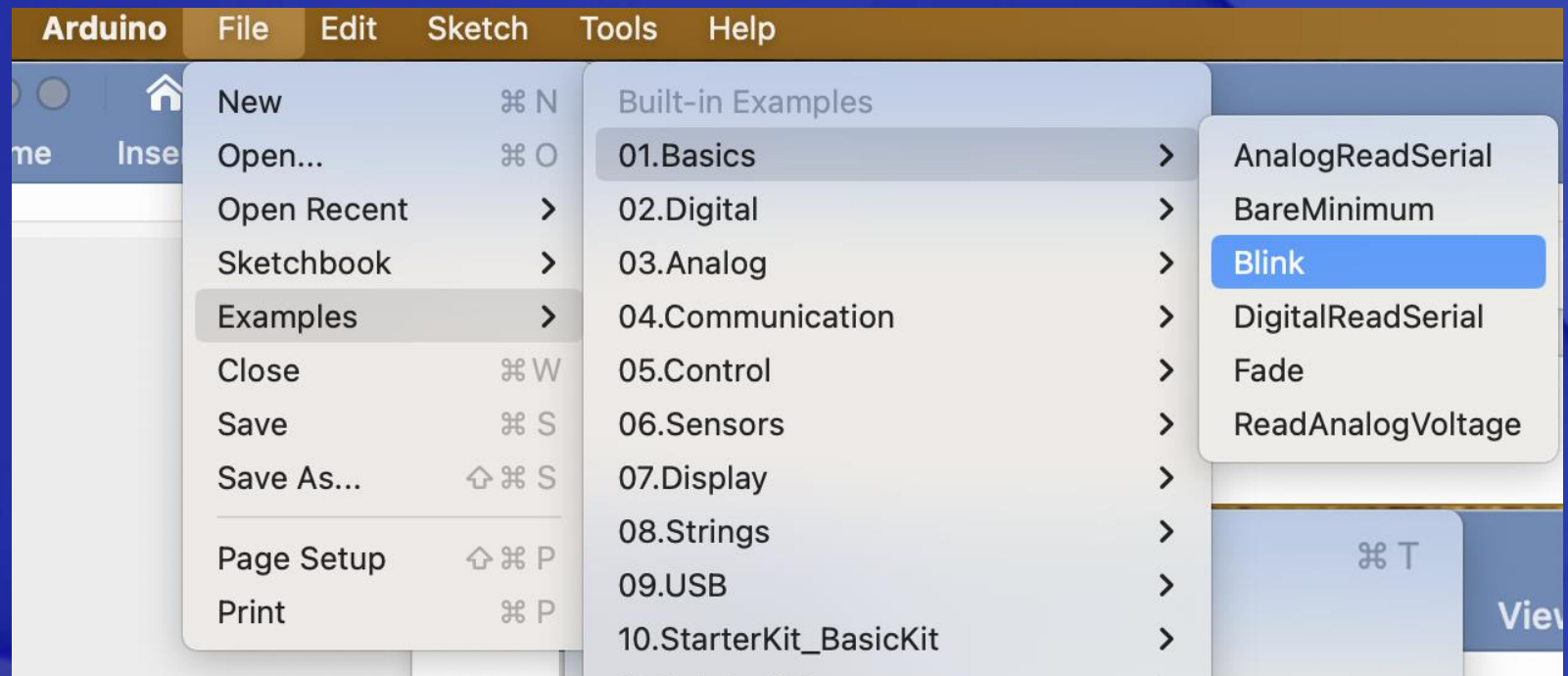
4. Arduino: Assign Serial Port

5. Arduino: Assign Board Type

Board is: NodeMCU (ESP-12E)



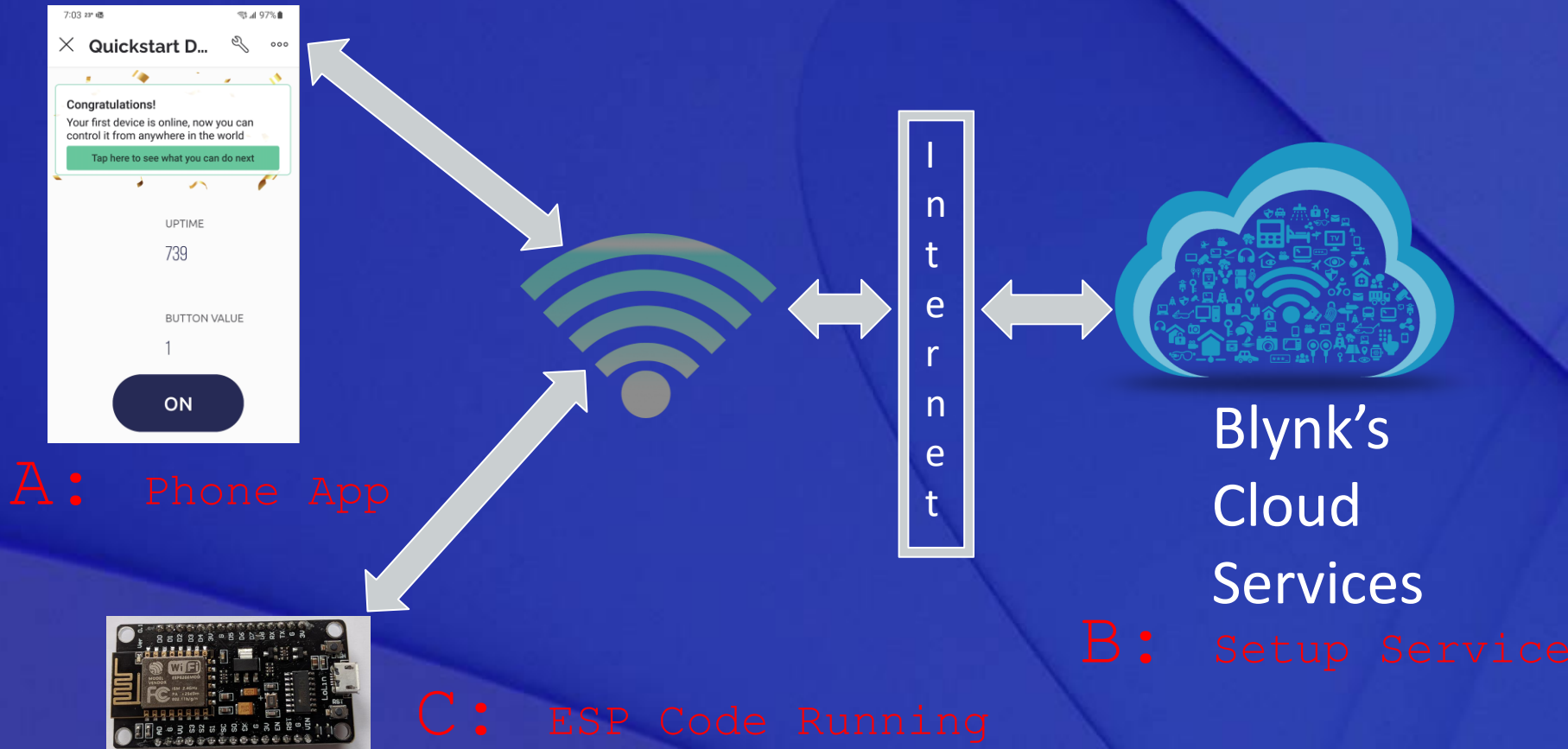
6. Install/Run Blink Demo



Compile/Upload/Run Blue built-in light blinks

Lab-10 Part 2 1/8

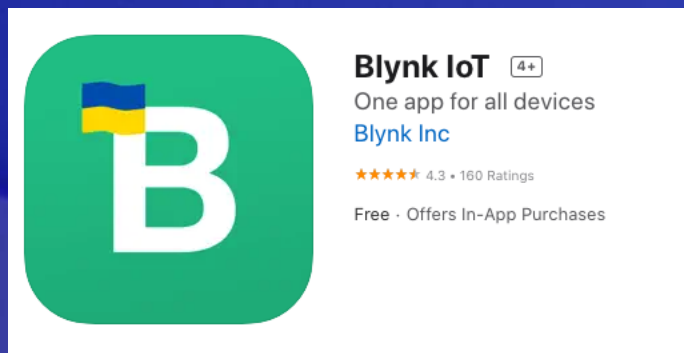
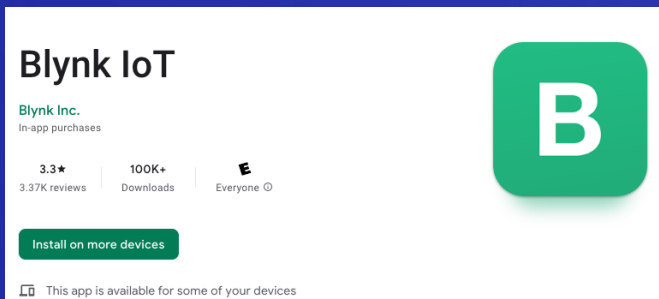
This part connects ESP8266 to Cloud Service



Lab-10 Part 2 2/8

To get connection pieces

1. Install Phone App:



A: Phone App

- B: Setup Services
- ## 2. Create Blynk Account (blynk.cloud) It's free verify your email, etc.

A screenshot of the Blynk Sign Up page. At the top is the Blynk logo (a green square with a white 'B'). Below it is the heading "Sign Up". The text says "Welcome! Fill in your email address and we will send an account activation link." There is an "EMAIL" input field with an envelope icon and a red "x" icon. Below the input field is a checkbox with the text "I agree to Terms and Conditions and accept Privacy Policy". At the bottom is a green "Sign Up" button and a blue "Back to Login" link.

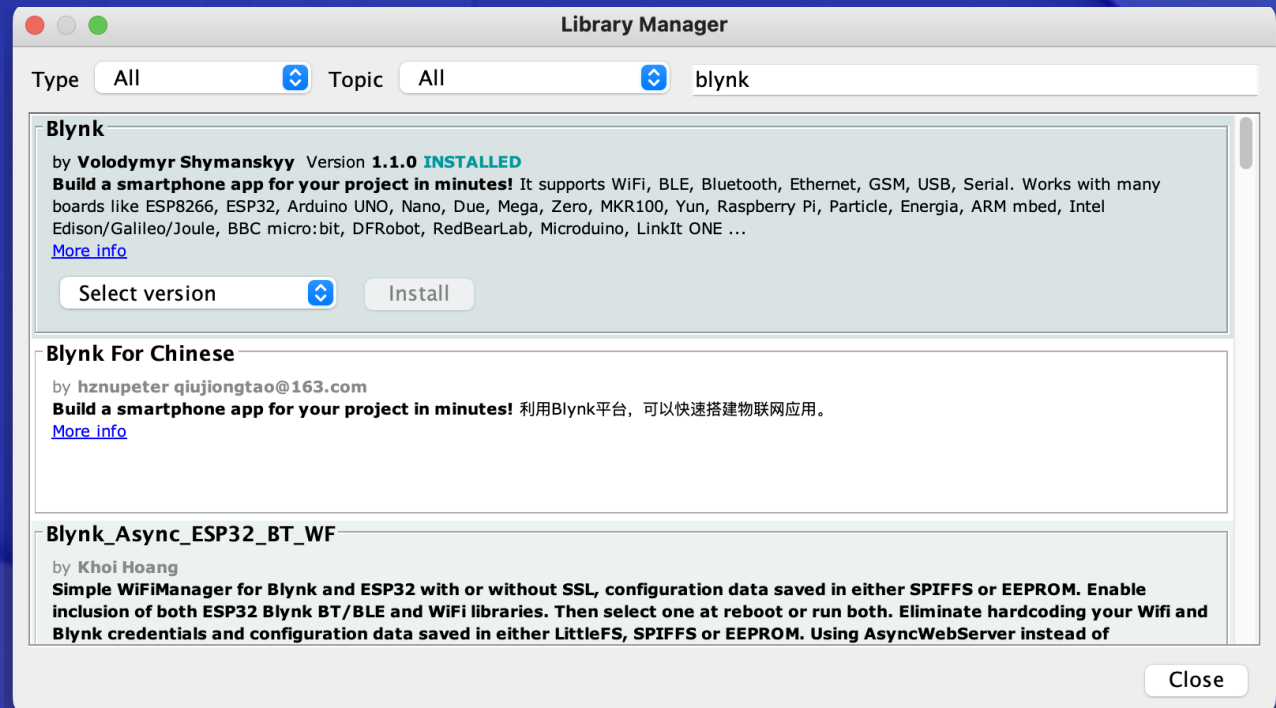
Lab-10 Part 2 3/8

C: ESP Code Running ([LINK HERE](#))

1. Install Blynk SDK into Arduino System

Arduino: Tools/Library Manager Search Blynk

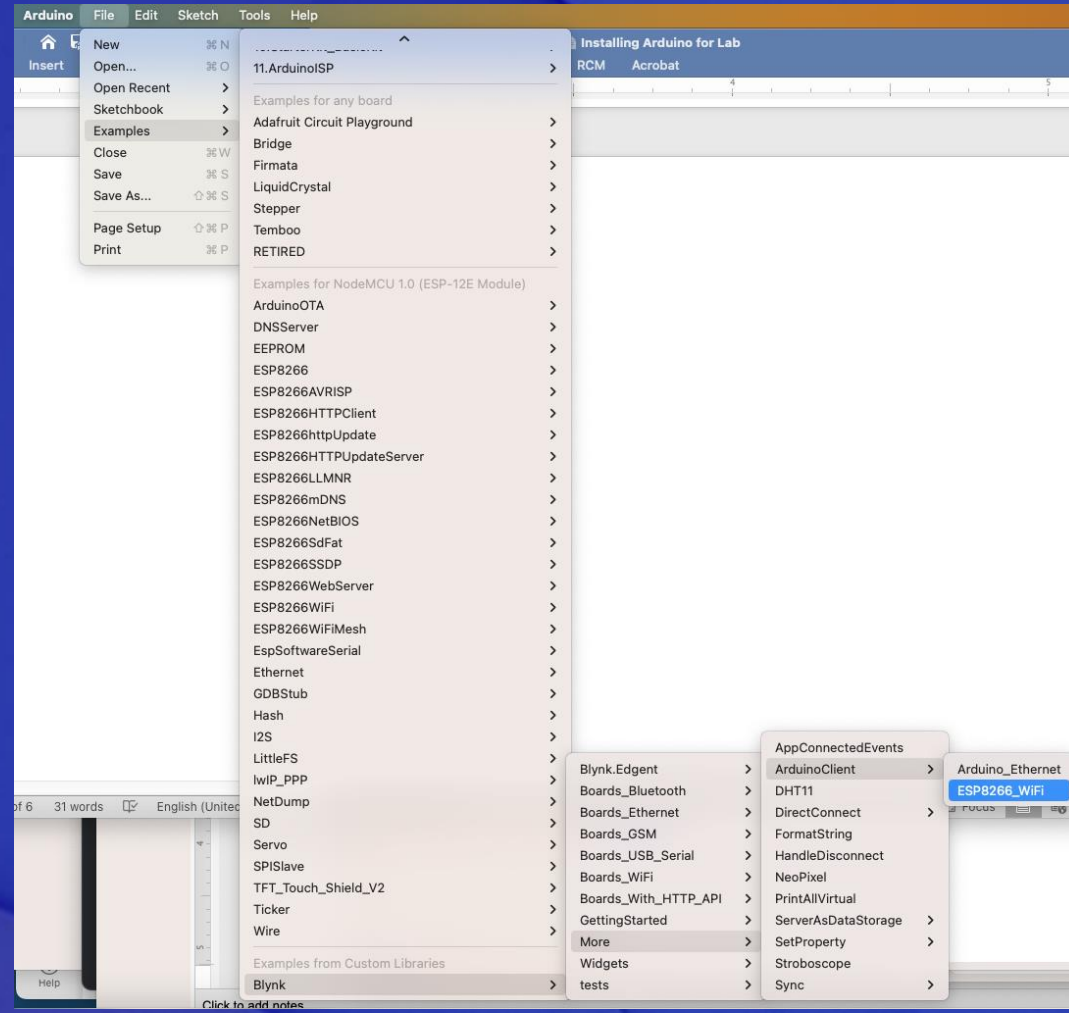
Install latest
version



Lab-10 Part 2 4/8

C: ESP Code Running

Create a new Example
Arduino Sketch
(not too important,
we'll replace code)

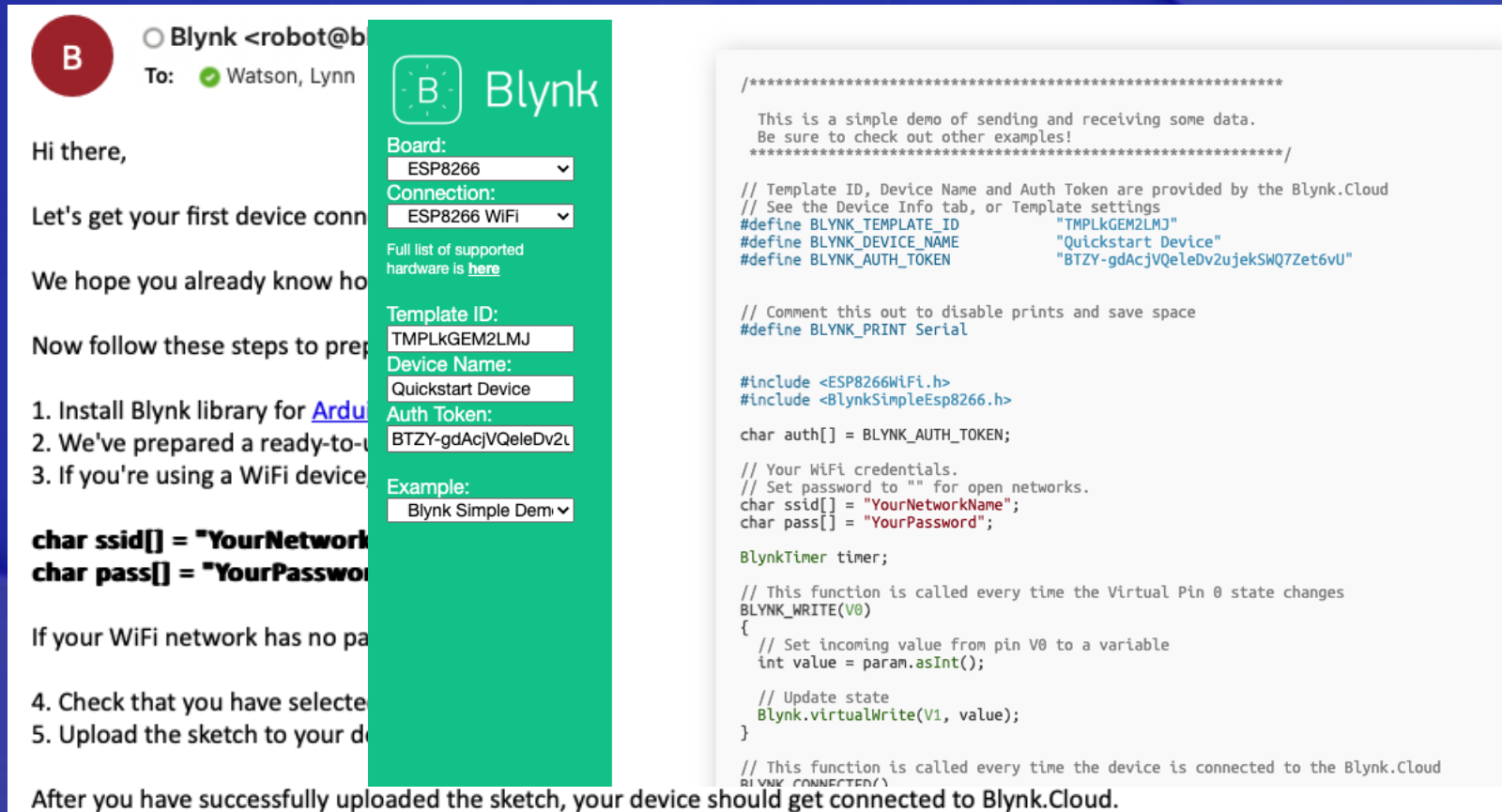


Lab-10 Part 2 5/8

C: ESP Code Running

Go to the email from blynk.cloud quickstart

Copy
Code
from
Link



The image shows a screenshot of an email from Blynk and a code editor. The email is from Blynk <robot@blynk.cloud> to Watson, Lynn. It contains instructions for setting up a Blynk device. The code editor shows the Blynk quickstart code for an ESP8266.

Email Content:

From: Blynk <robot@blynk.cloud>
To: Watson, Lynn

Hi there,

Let's get your first device connected.

We hope you already know how to connect a device to Blynk. Now follow these steps to prepare your device:

1. Install Blynk library for [Arduino](#)
2. We've prepared a ready-to-use code for you.
3. If you're using a WiFi device, you need to provide your WiFi credentials.

Code Snippets:

```
char ssid[] = "YourNetworkName";  
char pass[] = "YourPassword";
```

If your WiFi network has no password, you can use an empty string for the password.

4. Check that you have selected the correct board and connection.
5. Upload the sketch to your device.

After you have successfully uploaded the sketch, your device should get connected to Blynk.Cloud.

Blynk Configuration:

- Board: ESP8266
- Connection: ESP8266 WIFI
- Full list of supported hardware is [here](#)
- Template ID: TEMPLKEM2LMJ
- Device Name: Quickstart Device
- Auth Token: BTZY-gdAcjVQeLeDv2uJekSHQ7Zet6vU
- Example: Blynk Simple Demo

Code Editor Content:

```
/*  
 * This is a simple demo of sending and receiving some data.  
 * Be sure to check out other examples!  
 */  
  
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud  
// See the Device Info tab, or Template settings  
#define BLYNK_TEMPLATE_ID "TEMPLKEM2LMJ"  
#define BLYNK_DEVICE_NAME "Quickstart Device"  
#define BLYNK_AUTH_TOKEN "BTZY-gdAcjVQeLeDv2uJekSHQ7Zet6vU"  
  
// Comment this out to disable prints and save space  
#define BLYNK_PRINT Serial  
  
#include <ESP8266WiFi.h>  
#include <BlynkSimpleEsp8266.h>  
  
char auth[] = BLYNK_AUTH_TOKEN;  
  
// Your WiFi credentials.  
// Set password to "" for open networks.  
char ssid[] = "YourNetworkName";  
char pass[] = "YourPassword";  
  
BlynkTimer timer;  
  
// This function is called every time the Virtual Pin 0 state changes  
BLYNK_WRITE(V0)  
{  
  // Set incoming value from pin V0 to a variable  
  int value = param.asInt();  
  
  // Update state  
  Blynk.virtualWrite(V1, value);  
}  
  
// This function is called every time the device is connected to the Blynk.Cloud  
BLYNK_CONNECTED()  
{  
  // Set initial state  
  Blynk.virtualWrite(V1, 0);  
}
```

Lab-10 Part 2 5/8

C: ESP Code Running

Go to the email from blynk.cloud quickstart

Copy
Code
from
Link

Paste
Into
SRC .c

The image shows a screenshot of an email from Blynk and a code snippet for ESP8266. The email is from Blynk <robot@blynk.cloud> to Watson, Lynn. The subject is "Quickstart". The email body contains instructions for setting up a Blynk device. A green box highlights the Blynk logo and the following information:

- Board: ESP8266
- Connection: ESP8266 WIFI
- Full list of supported hardware is [here](#)
- Template ID: TMPLKGEM2LMJ
- Device Name: Quickstart Device
- Auth Token: BTZY-gdAcjVQeleDv2u
- Example: Blynk Simple Demo

The code snippet is a C++ program for an ESP8266. It includes the Blynk library and sets up the device name, auth token, and WiFi credentials. The code is as follows:

```
*****
This is a simple demo of sending and receiving some data.
Be sure to check out other examples!
*****

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPLKGEM2LMJ"
#define BLYNK_DEVICE_NAME "Quickstart Device"
#define BLYNK_AUTH_TOKEN "BTZY-gdAcjVQeleDv2u"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
    // Set incoming value from pin V0 to a variable
    int value = param.asInt();

    // Update state
    Blynk.virtualWrite(V1, value);
}

// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED()
{
    // Set device name and auth token
    Blynk.config(auth, ssid, pass);
}
```

Lab-10 Part 2 6/8

C: ESP Code Running

Modify Arduino Source to include your codes:

ECEN_361-Lab10-Blynk-Works-ESP8266_Standalone

```
/*
*****

This is a simple demo of sending and receiving some data.
Be sure to check out other examples!
*****

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID          "TMPLkGEM2[REDACTED]"
#define BLYNK_DEVICE_NAME          "Quickstart Device"
#define BLYNK_AUTH_TOKEN           "BTZY-gdAcjVQeleDvZujekSWQ[REDACTED]5vU"
```

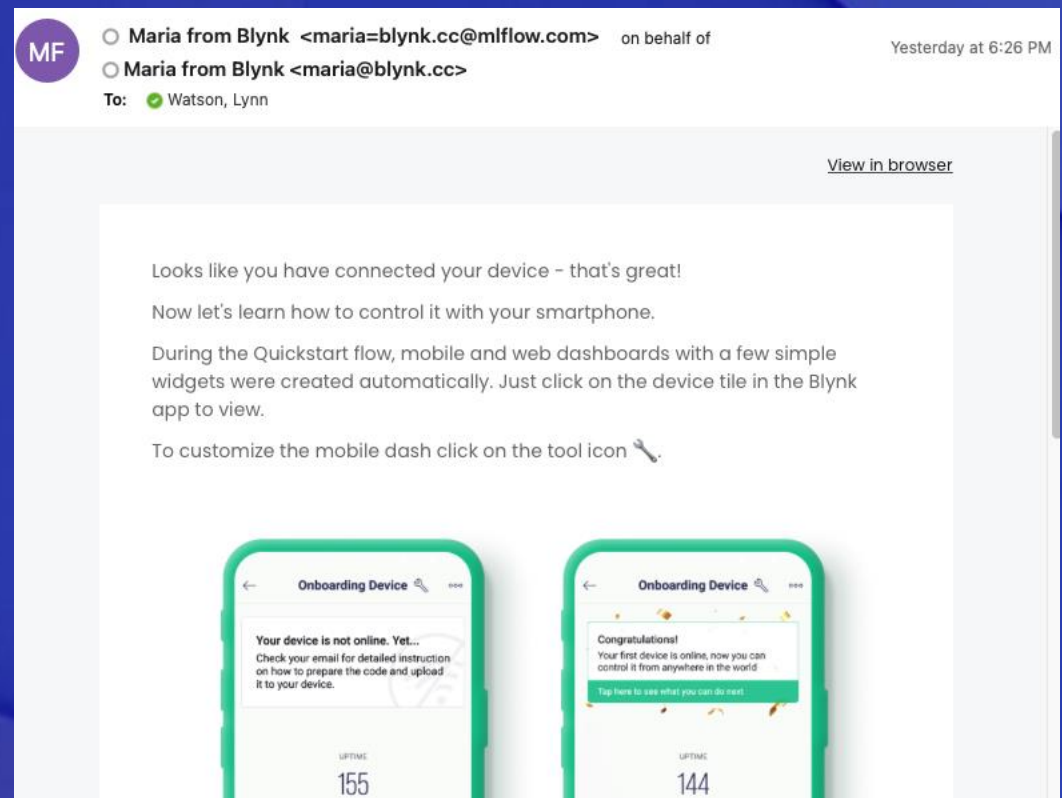
And WiFi:

```
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "BYU_Visitor";
char pass[] = "";
```


Lab-10 Part 2 7/8

C: ESP Code Running

- Run the code. If connected, you'll get an email!
- Doesn't do the LED yet! Just shows that you pressed the App button



Lab-10 Part 2 8/8

C: ESP Code Running

Connect the on-board LED to do something when the switch is pressed

- Set the onboard LED port direction

```
void setup()
{
  // Debug console
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT); // ECEN-361 ADDED
```

- Turn the light on/off based on the value from the cloud

```
// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
  // Set incoming value from pin V0 to a variable
  int value = param.asInt();

  digitalWrite(LED_BUILTIN, !value); // ECEN-361 ADDED -- LED tied high, so negative-true logic.
```