

ECEN 361

Real-Time and Embedded

Systems

Lab-10 Setup Description

ESP8266 Board and Cloud IOT

BRIGHAM YOUNG
UNIVERSITY
IDAH



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

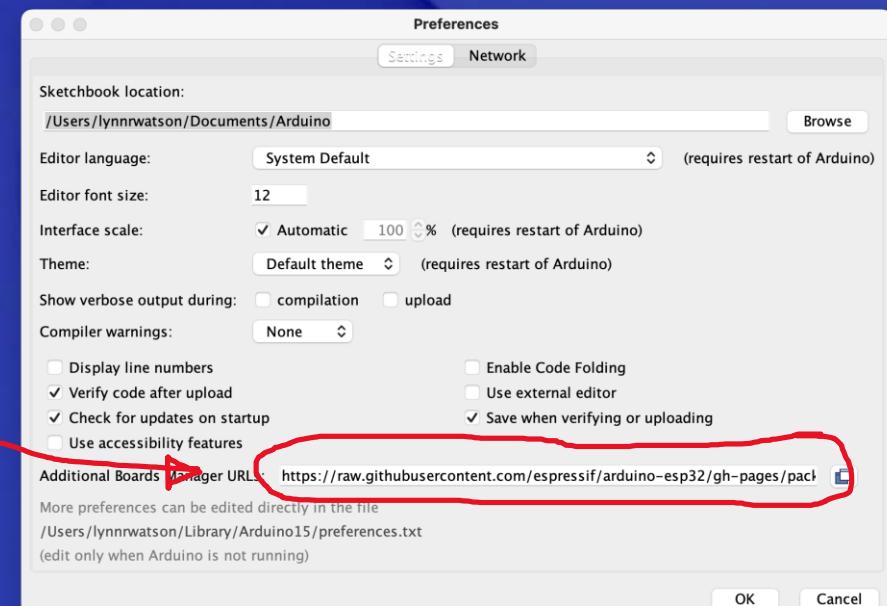
Steps: (Tutorial Online [HERE](#))

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

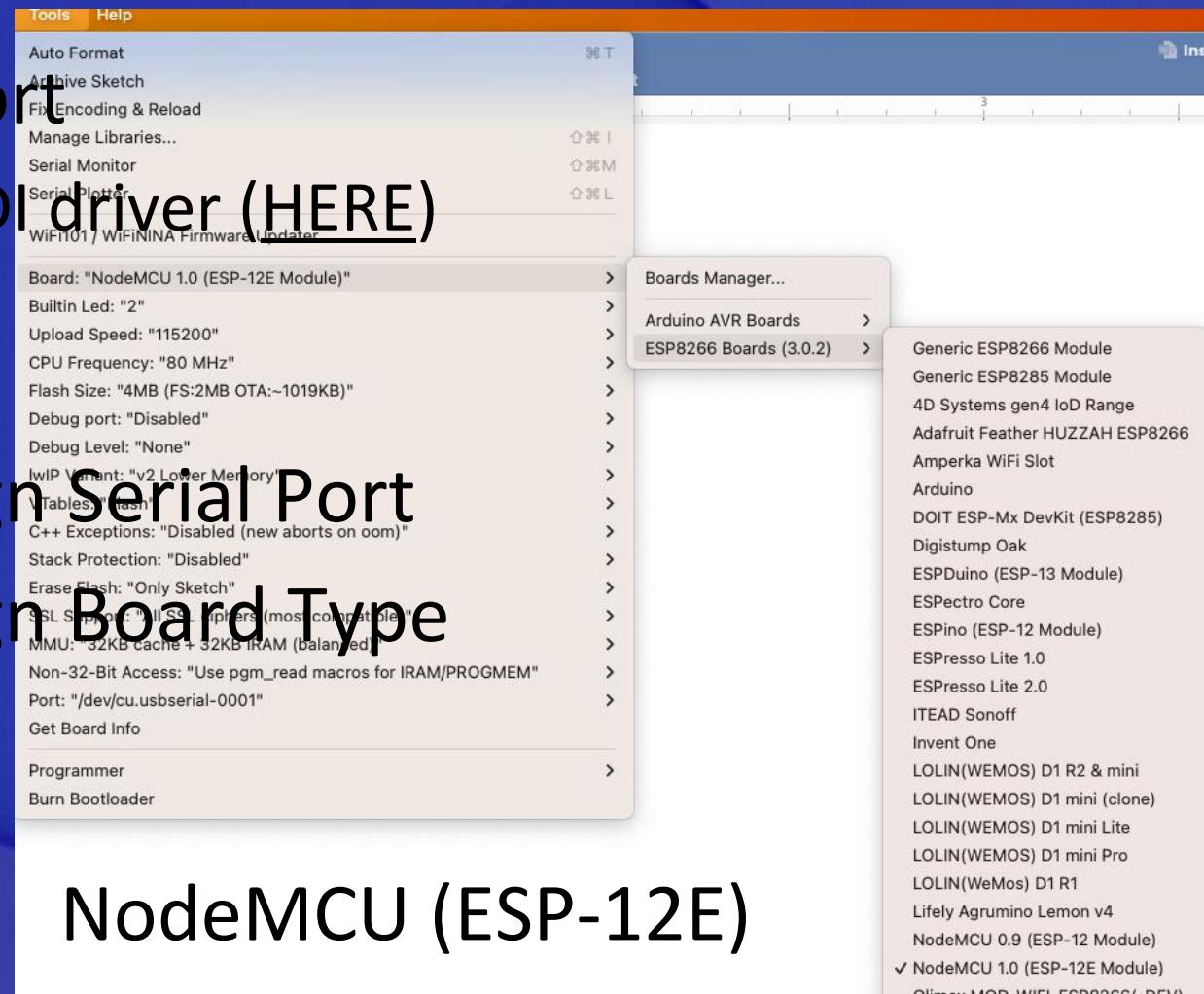
If not – Install FTDI driver ([HERE](#))

4. Arduino: Assign Serial Port

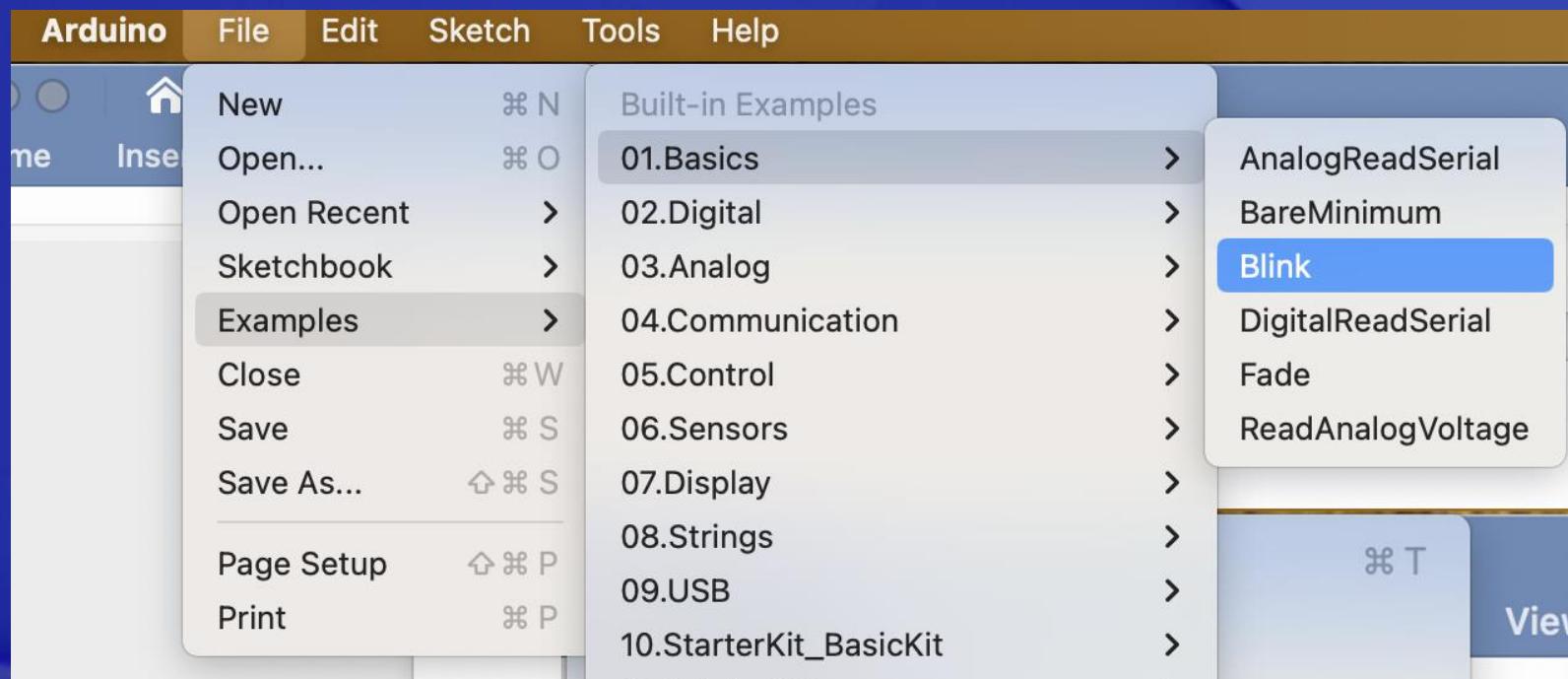
5. Arduino: Assign Board Type

Board is:

NodeMCU (ESP-12E)



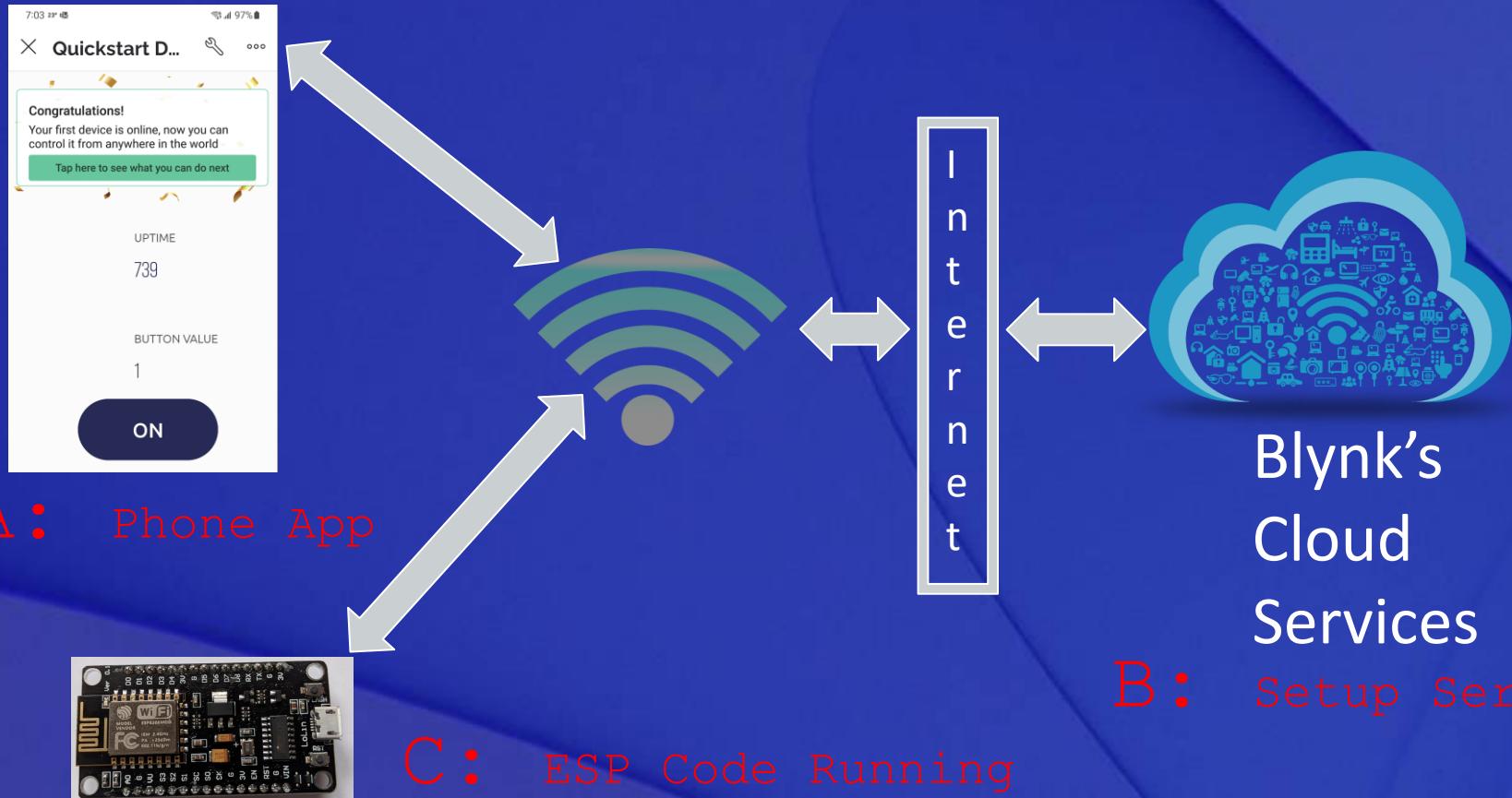
6. Install/Run Blink Demo



Compile/Upload/Run Blue built-in light blinks

Lab-10 Part 2 1/8

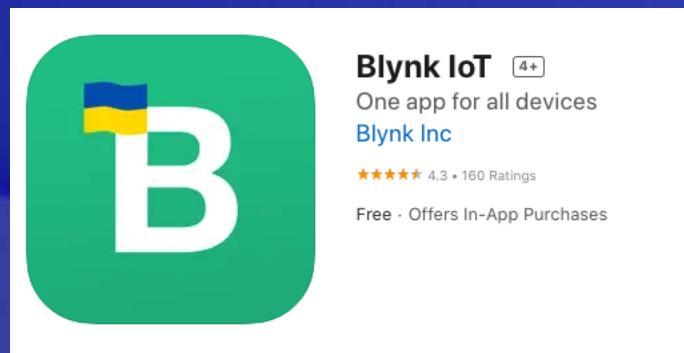
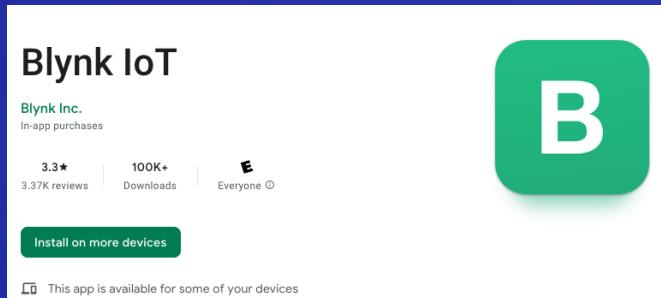
This part connects ESP8266 to Cloud Service



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.

The image shows the 'Sign Up' page for the Blynk account creation process. It has a large green 'B' icon at the top. Below it, the heading 'Sign Up' is centered. A welcome message reads: 'Welcome! Fill in your email address and we will send an account activation link.' There is a 'EMAIL' input field with a red '...' placeholder. Below it is a checkbox labeled 'I agree to Terms and Conditions and accept Privacy Policy'. At the bottom right is a green 'Sign Up' button, and at the very bottom center is a 'Back to Login' link.

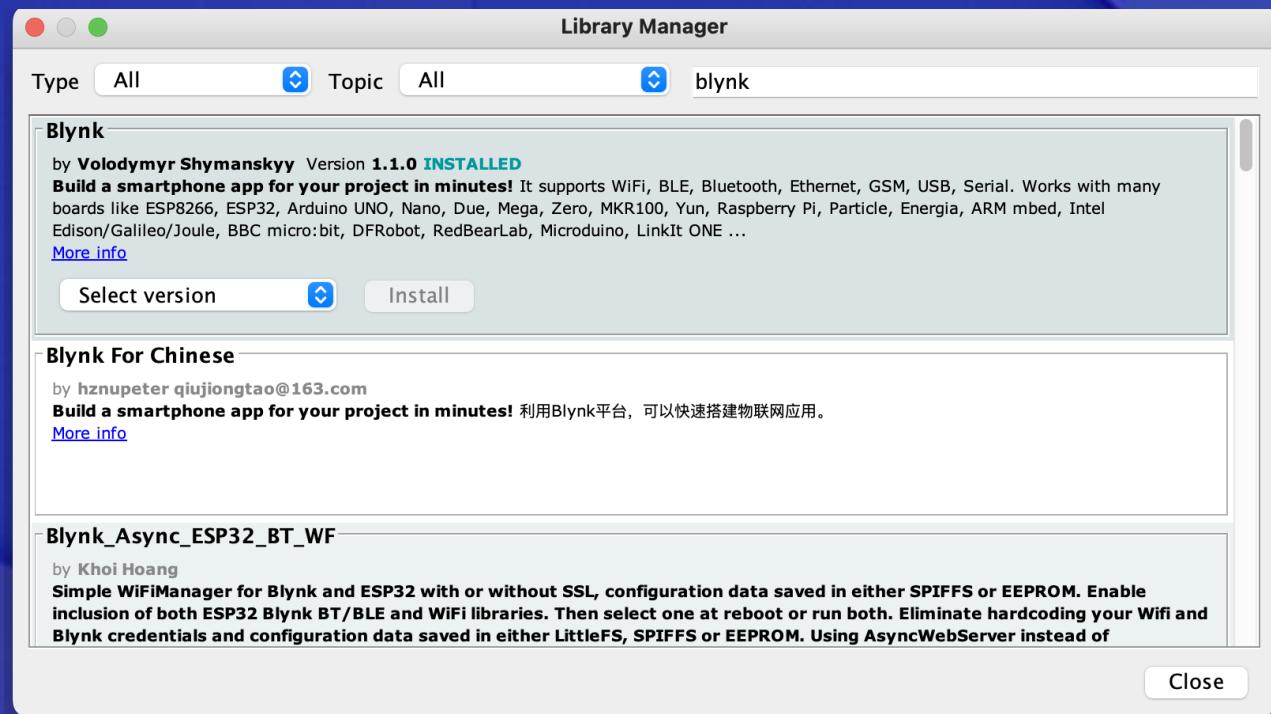
Lab-10 Part 2 3/8

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

1. Install Blynk SDK into Arduino System

Arduino: Tools/Library Manager Search Blynk

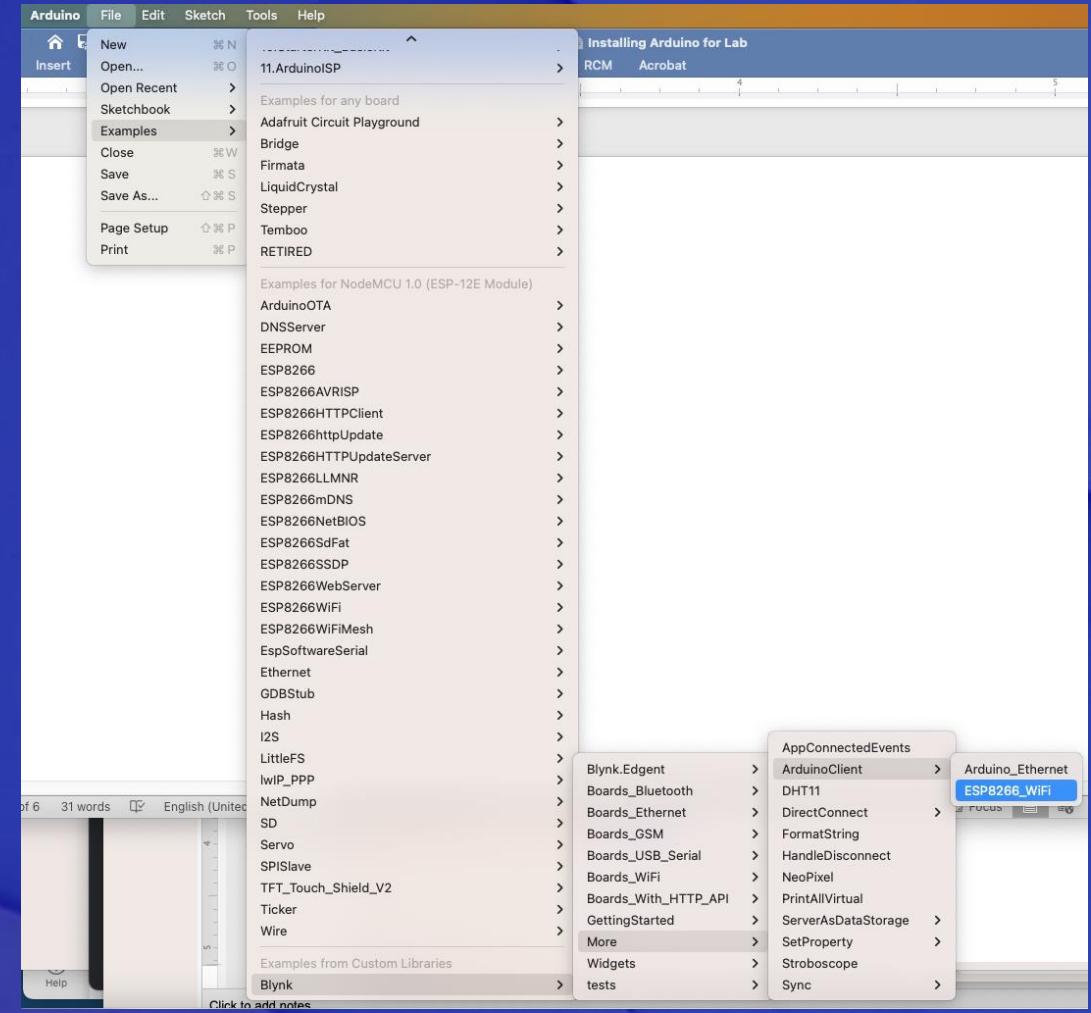
Install latest
version



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 two side-by-side screenshots. On the left is an email from Blynk to Watson, Lynn, with a subject line starting with 'B'. The email body contains instructions for connecting a device, mentioning steps like installing the Blynk library and providing WiFi credentials. On the right is a screenshot of the Blynk Cloud interface for an ESP8266 device named 'Quickstart Device' with auth token 'BTZY-gdAcjVQeleDv2ujekSWQ7Zet6vU'. It shows template ID 'TMPLKGEMLMJ', device name 'Quickstart Device', and auth token 'BTZY-gdAcjVQeleDv2ujekSWQ7Zet6vU'. Below these are sections for 'Example' (containing 'Blynk Simple Demo') and 'Code' (containing the following C code).

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

If your WiFi network has no password, skip to step 4.

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

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

```
/*****************************************************************************  
 * 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          "TMPLKGEMLMJ"  
#define BLYNK_DEVICE_NAME          "Quickstart Device"  
#define BLYNK_AUTH_TOKEN           "BTZY-gdAcjVQeleDv2ujekSWQ7Zet6vU"  
  
// 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()
```

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 two side-by-side screenshots. On the left is an email from Blynk to Watson, Lynn, with a subject line starting with 'B'. The email body says 'Hi there,' and 'Let's get your first device conn...'. It also includes a link to hardware support. On the right is a screenshot of the Blynk Cloud interface for an ESP8266 WiFi device. It shows the board selection as 'ESP8266', connection as 'ESP8266 WiFi', and template ID as 'TMPLKGEM2LMJ'. The device name is 'Quickstart Device' and the auth token is 'BTZY-gdAcjVQeleDv2ujekSWQ7Zet6vU'. An example sketch is shown at the bottom.

```
/*****************************************************************************  
 * 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-gdAcjVQeleDv2ujekSWQ7Zet6vU"  
  
 // 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  
 #ifdef BLYNK_CONNECTED()  
 /*****  
 *****/
```

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

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          "TMPLkGEMZLUD"
#define BLYNK_DEVICE_NAME          "Quickstart Device"
#define BLYNK_AUTH_TOKEN           "BTZY-gdAcjVQeleDv2ujekSWQZzvU"
```

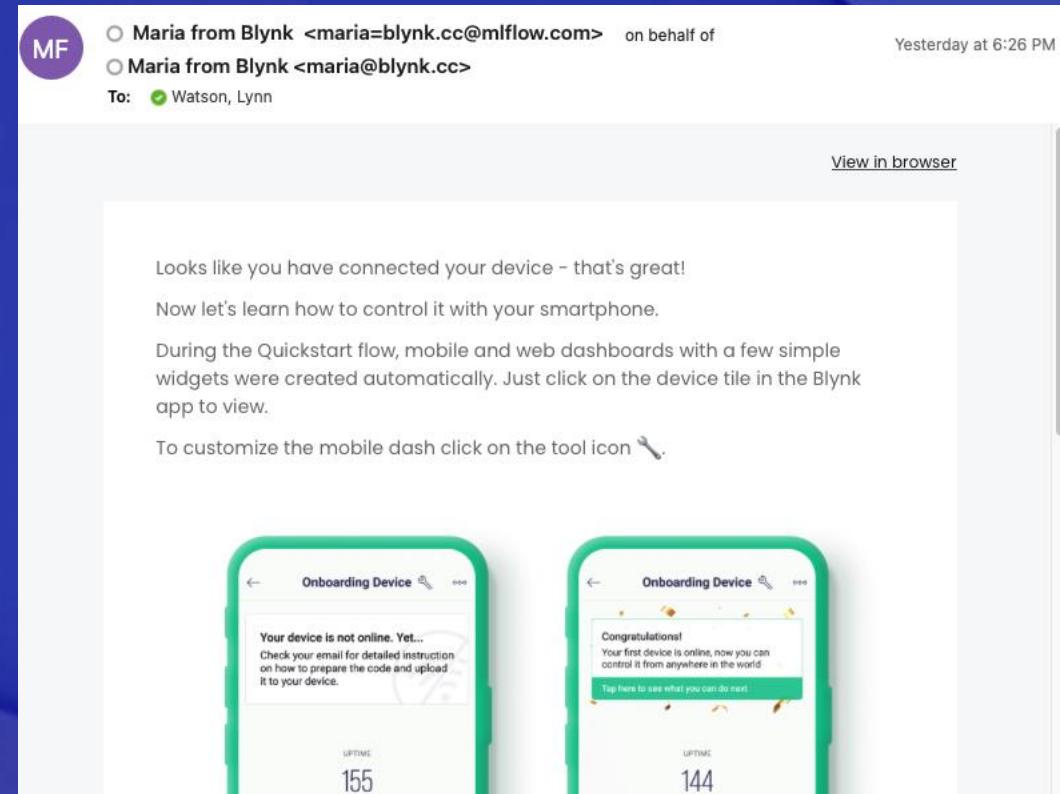
And WiFi:

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

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

```
// 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.toInt();

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