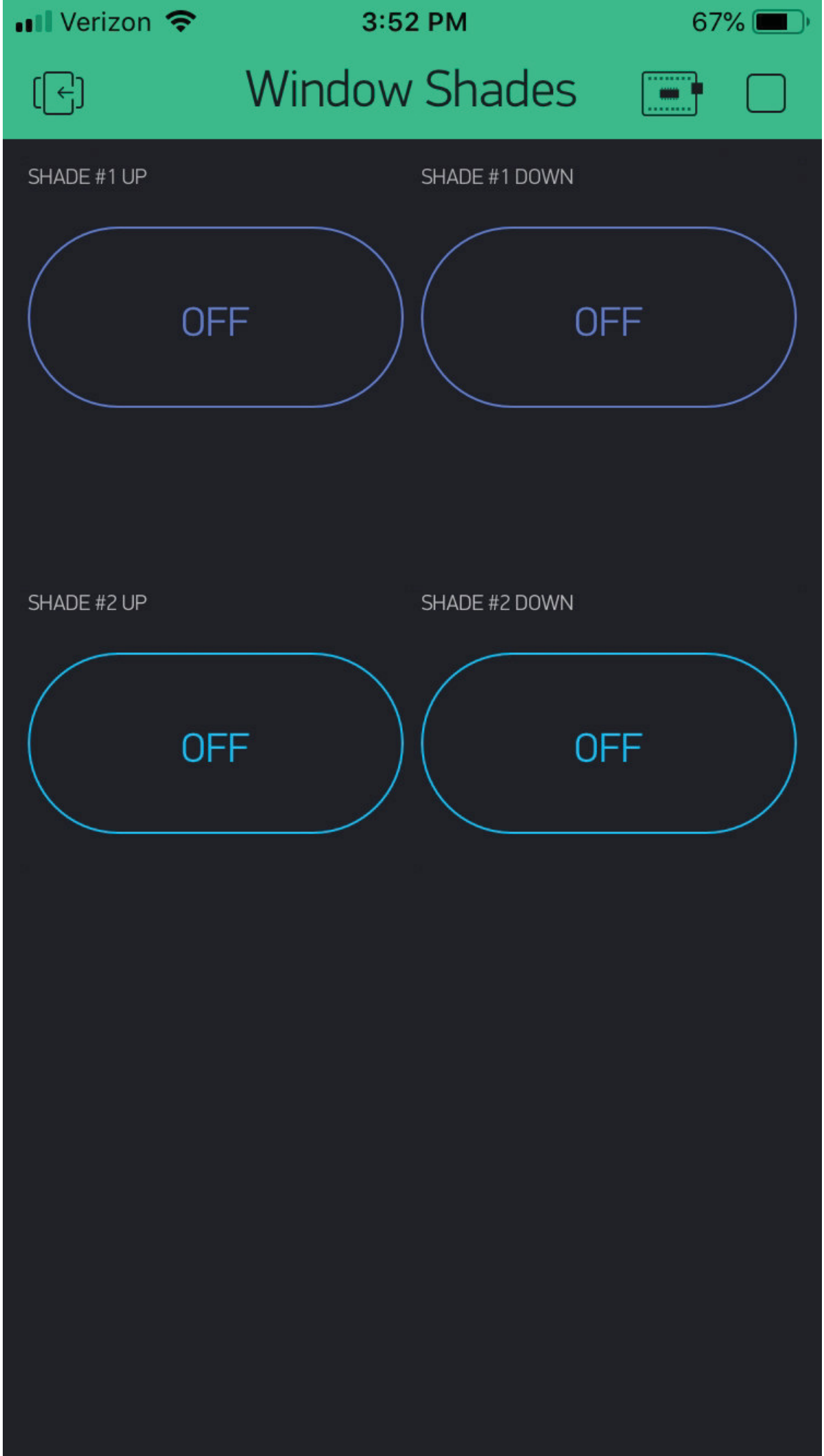
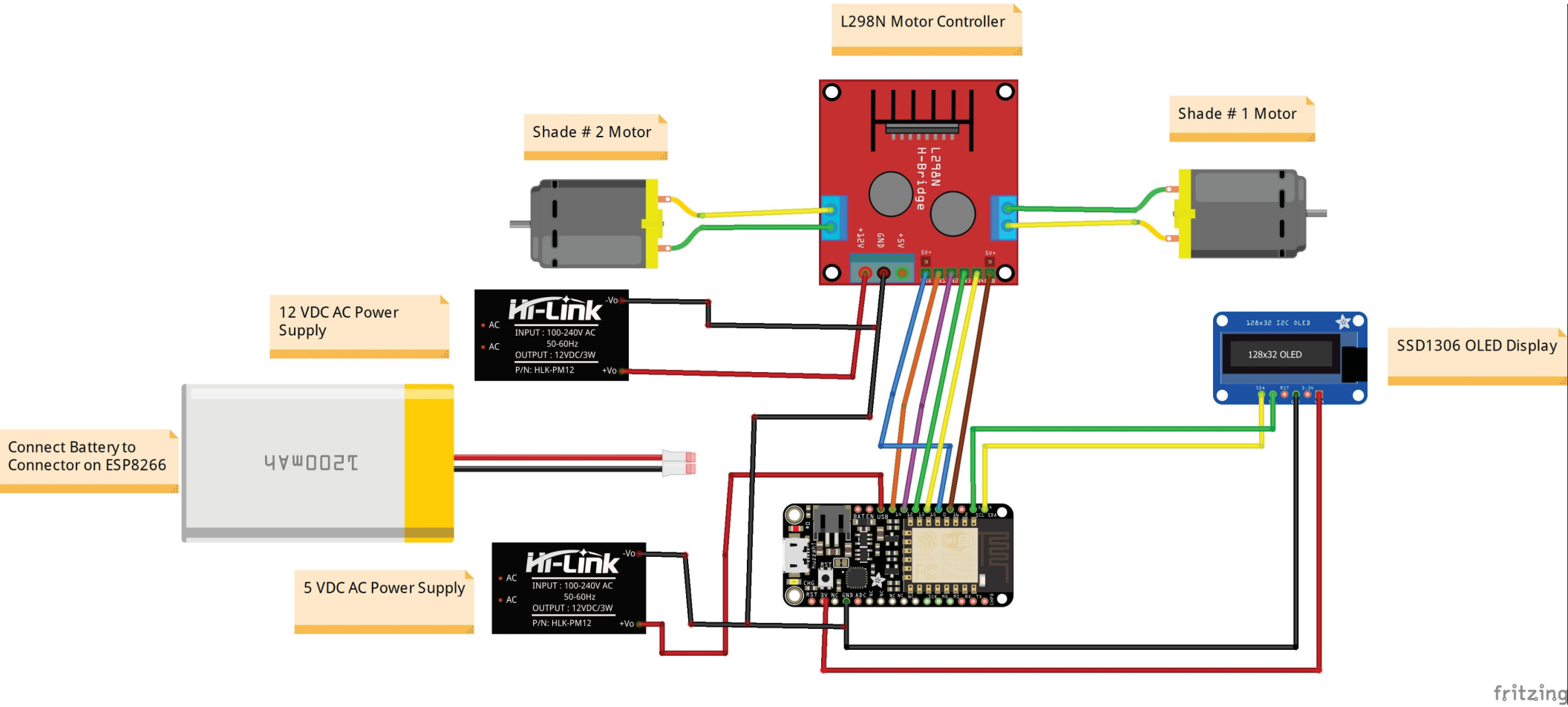

Verizon
3:52 PM
67%
Window Shades
SHADE #1 UP
SHADE #1 DOWN
OFF
OFF
SHADE #2 UP
SHADE #2 DOWN
OFF
OFF


L298N Motor Controller
Shade # 2 Motor
Shade # 1 Motor
L298N
H-Bridge
12 VDC AC Power Supply
Hi-Link
AC
AC
INPUT : 100-240V AC
50-60Hz
OUTPUT : 12VDC/3W
P/N: HLK-PM12
-Vo
+Vo
128x32 OLED
SSD1306 OLED Display
Connect Battery to Connector on ESP8266
5 VDC AC Power Supply
fritzing

```
/*
ESP8266 Program to Roll Up/Down two window shades:

Hardware Required:
-----------------
- Adafruit ESP8266 Feather
- Adafruit OLED Feather Wing Display Module using IIC protocol

The circuit has the following features:
 0) Starts as an Access Point, and connects to network via a phone or laptop through a GUI.
 1) Connects and Transmits / Recieves through the house wireless router as a client server.
 2) Connects to the Blynk cloud server webpage that could be pulled up anywhere.
 3) The "blue LED" lights when connected to Wi-Fi.
 4) Auto syncs back to last values on server when power is lost.

Note(s):
 * First download the Blynk App on your phone, and set up your account and Project.
 * Don't put Blynk.virtualWrite and any other Blynk.* command inside void loop() the connection will be terminated.
 * Call functions with intervals. For example, this SimpleTimer Library is a library for timed events.
 * Avoid using long delays with delay() - it may cause connection breaks;
 * If you send more than 100 values per second - you may cause Flood Error and your hardware will be disconnected from the server.
 * Be careful sending a lot of Blynk.virtualWrite commands as most hardware is not very powerful (like ESP8266) so it may not handle many requests.
 * When first connecting to a network, go to Wi-Fi settings when already connected to the house router, and choose "Window Shade" to set up.

OLED Wiring
-----------
 * VDD  = + 3v of Feather Wing
 * Gnd = Gnd of Feather Wing
 * SDA of Feather Wing
 * SCL of Feather Wing
*/
// Define Libraries
// ================
#include <Adafruit_SSD1306.h>
#include "Adafruit_GFX.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <Wire.h>
#include <SPI.h>
#include "Adafruit_LEDBackpack.h"
#include <Adafruit_SSD1306.h>
#include "Adafruit_GFX.h"
#include <BlynkSimpleEsp8266.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>
#include <SimpleTimer.h>

// L298N Motor Controller Wiring
// -----------------------------
// Shade Motor One
// ---------------
#define enA 0
#define in1 14
#define in2 12

// Shade Motor Two
// ---------------
#define enB 16
#define in3 13
#define in4 15
```

```
63
64  // OLED Screen Setup
65  // -----------------
66  #define SCREEN_WIDTH 128 // OLED display width, in pixels
67  #define SCREEN_HEIGHT 32 // OLED display height, in pixels
68  // Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
69  #define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
70  Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
71
72
73  // Declare Global Variables
74  // ========================
75  char auth[] = "[redacted]";  // Put your Auth Token here.
76
77  SimpleTimer timer; // Create a Timer case
78
79  BLYNK_WRITE(V0){  // Read virtual push button for Shade #1 UP
80    int shade1up = param.asInt(); // Create a local variable to store virtual button value
81      if (shade1up == 1){  // Check to see if virtual button was depressed to "on"
82          Shade1_UP(); // Goto Function
83      }
84      else {
85         Shade1_OFF(); // Goto Function
86      }
87  }
88
89  BLYNK_WRITE(V1){  // Read virtual push button for Shade #1 Down
90    int shade1down = param.asInt(); // Create a local variable to store virtual button
      value
91      if (shade1down == 1){  // Check to see if virtual button was depressed to "on"
92          Shade1_DN(); // Goto Function
93      }
94      else {
95         Shade1_OFF(); // Goto Function
96      }
97  }
98
99  BLYNK_WRITE(V2){  // Read virtual push button for Shade #2 UP
100   int shade2up = param.asInt(); // Create a local variable to store virtual button value
101     if (shade2up == 1){  // Check to see if virtual button was depressed to "on"
102         Shade2_UP(); // Goto Function
103     }
104     else {
105        Shade2_OFF(); // Goto Function
106     }
107 }
108
109 BLYNK_WRITE(V3){  // Read virtual push button for Shade #2 Down
110   int shade2down = param.asInt(); // Create a local variable to store virtual button
      value
111     if (shade2down == 1){  // Check to see if virtual button was depressed to "on"
112         Shade2_DN(); // Goto Function
113     }
114     else {
115        Shade2_OFF(); // Goto Function
116     }
117 }
118
119 // Main Program
120 // ============
121 void setup() {
122   WiFiManager wifi;  // Start an AP, and a GUI to enter Wi-Fi information if one
      already does not exist
123   //wifi.resetSettings(); // Uncomment to reset saved wi-fi settings
124   wifi.autoConnect("Window_Shade");
125   Blynk.config(auth); // Make the cloud server connection
126   delay(1000); // Allow board to settle, and enough time to enter "boot mode"
127   Serial.begin(115200);
128   // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
```

```
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }
 display.clearDisplay();
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(0, OUTPUT); // Set red LED as an output to indicate sampling sensor for data
  pinMode(2, OUTPUT); // Set blue LED as an output to indicate satisfactory Wi-Fi
  connection
  digitalWrite(2, HIGH);  // Turn off blue LED
  while(WiFi.status() != WL_CONNECTED) {
    digitalWrite(2, HIGH);  // Turn off blue LED
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
  digitalWrite(2, LOW);  // Turn on BLUE LED to show Wi-Fi connection
  display.ssd1306_command(SSD1306_DISPLAYON); // Switch display back on, if it has shut
  off
  display.clearDisplay();  // Clear buffer
  display.setTextSize(1); // Set OLED text size ("1" provides 4 lines of data about 20
  characters per line)
  // ("2" provides 2 lines of data about 10 characters per line)
  display.setTextColor(WHITE);  // Sets color
  display.setCursor(0,0); // Set cursor to line 1, position zero, use (0,16) to go to
  second line, etc.
  display.print("Connected");
  //  display.println(ssid);
  display.setCursor(0,16); // Set cursor to line 3, position zero, use (0,16) to go to
  3rd line with size = 1.
  display.print("IP Address: ");
  display.println(WiFi.localIP()); // Display IP Address
  display.display();  // Write to Display Buffer
  delay(2000); // 2 second delay
  display.clearDisplay();  // Clear buffer
  display.setTextSize(1); // Set OLED text size ("1" provides 4 lines of data about 20
  characters per line)
  // ("2" provides 2 lines of data about 10 characters per line)
  display.setTextColor(WHITE);  // Sets color
  display.setCursor(0,0); // Set cursor to line 1
  display.println("Window Shade Control");  // Display Welcome message
  display.println();
  display.println("By Roy H. Guerra Jr");
  display.display();  // Write to Display Buffer
  delay(2000); // 2 second delay
  timer.setInterval(300000L, blankscreen); // Setup a function to be called every 5
  minutes
}

BLYNK_CONNECTED() {
  Blynk.syncAll(); // Sync all virtual devices to last values stored on server.
}

void loop() {
  Blynk.run();
  timer.run();
}

void blankscreen(){
  display.clearDisplay();
  display.ssd1306_command(SSD1306_DISPLAYOFF); // Switch display off
}

void Shade1_UP(){ // This function will pull up Shade #1
```

```
191     display.ssd1306_command(SSD1306_DISPLAYON); // Switch display back on, if it has shut
        off
192     digitalWrite(in1, HIGH); // turn on motor A (forward)
193     digitalWrite(in2, LOW);
194     digitalWrite(enA, HIGH); // set speed to Full
195     Serial.println("Motor A Forward");
196     display.clearDisplay();  // Clear buffer
197     display.setTextSize(1); // Set OLED text size ("1" provides 4 lines of data about 20
        characters per line)
198     // ("2" provides 2 lines of data about 10 characters per line)
199     display.setTextColor(WHITE);  // Sets color
200     display.setCursor(0,0); // Set cursor to line 1
201     display.println("Window Shade #1");  // Display Welcome message
202     display.println();
203     display.println("Direction = Up");
204     display.display();  // Write to Display Buffer
205   }
206
207   void Shade1_DN(){ // This function will pull down Shade #1
208     display.ssd1306_command(SSD1306_DISPLAYON); // Switch display back on, if it has shut
        off
209     digitalWrite(in1, LOW); // turn on motor A (reverse)
210     digitalWrite(in2, HIGH);
211     digitalWrite(enA, HIGH); // set speed to Full
212     Serial.println("Motor A Reverse");
213     display.clearDisplay();  // Clear buffer
214     display.setTextSize(1); // Set OLED text size ("1" provides 4 lines of data about 20
        characters per line)
215     // ("2" provides 2 lines of data about 10 characters per line)
216     display.setTextColor(WHITE);  // Sets color
217     display.setCursor(0,0); // Set cursor to line 1
218     display.println("Window Shade #1");  // Display Welcome message
219     display.println();
220     display.println("Direction = Down");
221     display.display();  // Write to Display Buffer
222   }
223
224   void Shade1_OFF(){ // This function will stop Shade #1 Movement
225     display.ssd1306_command(SSD1306_DISPLAYON); // Switch display back on, if it has shut
        off
226     digitalWrite(in1, LOW); // turn on motor A (stopped)
227     digitalWrite(in2, LOW);
228     digitalWrite(enA, LOW); // set speed to 0
229     Serial.println("Motor A Stopped");
230     display.clearDisplay();  // Clear buffer
231     display.setTextSize(1); // Set OLED text size ("1" provides 4 lines of data about 20
        characters per line)
232     // ("2" provides 2 lines of data about 10 characters per line)
233     display.setTextColor(WHITE);  // Sets color
234     display.setCursor(0,0); // Set cursor to line 1
235     display.println("Window Shade #1");  // Display Welcome message
236     display.println();
237     display.println("Direction = Stop");
238     display.display();  // Write to Display Buffer
239   }
240
241   void Shade2_UP(){ // This function will pull up Shade #2
242     display.ssd1306_command(SSD1306_DISPLAYON); // Switch display back on, if it has shut
        off
243     digitalWrite(in3, HIGH); // turn on motor B (forward)
244     digitalWrite(in4, LOW);
245     digitalWrite(enB, HIGH); // set speed to Full
246     Serial.println("Motor B Forward");
247     display.clearDisplay();  // Clear buffer
248     display.setTextSize(1); // Set OLED text size ("1" provides 4 lines of data about 20
        characters per line)
249     // ("2" provides 2 lines of data about 10 characters per line)
250     display.setTextColor(WHITE);  // Sets color
251     display.setCursor(0,0); // Set cursor to line 1
```

```
252     display.println("Window Shade #2");  // Display Welcome message
253     display.println();
254     display.println("Direction = Up");
255     display.display();  // Write to Display Buffer
256   }
257
258   void Shade2_DN(){ // This function will pull down Shade #2
259     display.ssd1306_command(SSD1306_DISPLAYON); // Switch display back on, if it has shut
      off
260     digitalWrite(in3, LOW); // turn on motor B (reverse)
261     digitalWrite(in4, HIGH);
262     digitalWrite(enB, HIGH); // set speed to Full
263     Serial.println("Motor B Reverse");
264     display.clearDisplay();  // Clear buffer
265     display.setTextSize(1); // Set OLED text size ("1" provides 4 lines of data about 20
      characters per line)
266     // ("2" provides 2 lines of data about 10 characters per line)
267     display.setTextColor(WHITE);  // Sets color
268     display.setCursor(0,0); // Set cursor to line 1
269     display.println("Window Shade #2");  // Display Welcome message
270     display.println();
271     display.println("Direction = Down");
272     display.display();  // Write to Display Buffer
273   }
274
275   void Shade2_OFF(){ // This function will stop Shade #1 Movement
276     display.ssd1306_command(SSD1306_DISPLAYON); // Switch display back on, if it has shut
      off
277     digitalWrite(in3, LOW); // turn on motor B (stopped)
278     digitalWrite(in4, LOW);
279     digitalWrite(enB, LOW); // set speed to 0
280     Serial.println("Motor B Stopped");
281     display.clearDisplay();  // Clear buffer
282     display.setTextSize(1); // Set OLED text size ("1" provides 4 lines of data about 20
      characters per line)
283     // ("2" provides 2 lines of data about 10 characters per line)
284     display.setTextColor(WHITE);  // Sets color
285     display.setCursor(0,0); // Set cursor to line 1
286     display.println("Window Shade #2");  // Display Welcome message
287     display.println();
288     display.println("Direction = Stop");
289     display.display();  // Write to Display Buffer
290   }
291
292
293
```