

```

// Fill-in information from your Blynk Template here
#define BLYNK_TEMPLATE_ID "TMPL141FDUzN"
#define BLYNK_DEVICE_NAME "ESP8266"
#define BLYNK_AUTH_TOKEN ""
#define BLYNK_FIRMWARE_VERSION "0.1.0"
#define BLYNK_PRINT Serial
//#define BLYNK_DEBUG
#define APP_DEBUG
// Uncomment your board, or configure a custom board in
Settings.h
//#define USE_SPARKFUN_BLYNK_BOARD
//#define USE_NODE_MCU_BOARD
//#define USE_WITTY_CLOUD_BOARD
//#define USE_WEMOS_D1_MINI
#include <Stepper.h>
#include "BlynkEdgent.h"
#include <NTPClient.h>
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
//#include <Servo.h> // including servo library.
const long utcOffsetInSeconds = 19800;
const int STEPS_PER_REVOLUTION = 2048;
const long ROTATION_RPM = 10;
const int ROTATION_DEGREES = 45;
// STEPS = STEPS_PER_REVOLUTION * ROTATION_DEGREES / 360
const int STEPS = 256;
// Creates an instance of stepper class
// Pins entered in sequence IN1-IN3-IN2-IN4 for proper step
sequence
Stepper stepper = Stepper(STEPS_PER_REVOLUTION, D5, D7, D6, D8);

bool Right = false;
bool Left = false;

WiFiUDP ntpUDP;

```

```
NTPClient timeClient(ntpUDP, "world.pool.ntp.org",
utcOffsetInSeconds);
int HH, MM, final_time;
int time_blynk, data;
BLYNK_WRITE(V0){
  Left = param.asInt();
}
```

```
BLYNK_WRITE(V1){
  Right = param.asInt();
}
//Servo servo_1;
#define motor D0
BLYNK_WRITE(V2)
{
  data = param.asInt();
}
BLYNK_WRITE(V3)
{
  time_blynk = param.asInt();
  Serial.print(time_blynk);
  //Serial.print("Got The Time");
}
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "";
char pass[] = "";
void setup()
{
  Serial.begin(115200);
  delay(100);
```

```
  BlynkEdgent.begin();
  pinMode(motor, OUTPUT);
  digitalWrite(motor, LOW);
  //servo_1.attach(0);
  timeClient.begin();
```

```
}
```

```
void loop() {  
  timeClient.update();  
  HH = timeClient.getHours();  
  //Serial.print(timeClient.getHours());  
  int hours = HH;  
  if (hours == 0) hours = 12; // Midnight  
  if (hours > 12) hours = hours - 12;  
  //Serial.print(hours);  
  //Serial.print(":");  
  MM = timeClient.getMinutes();  
  // Serial.println(timeClient.getMinutes());  
  final_time = 3600*hours + 60*MM;  
  // Serial.println(final_time);  
  if (time_blynk == final_time){  
    digitalWrite(motor, LOW);  
    delay(3000);  
    digitalWrite(motor, HIGH);  
  }  
  if (data == 1){  
    digitalWrite(motor, HIGH);  
    delay(5000);  
    digitalWrite(motor, LOW);  
    Serial.print("Opening");  
  }  
  delay(6000);  
  BlynkEdgent.run();  
  Blynk.run();  
  
  if (Right){  
    Serial.println("turning right");  
    stepper.setSpeed(ROTATION_RPM);  
    stepper.step(STEPS);  
  }  
  delay(20);  
}
```

```
if (Left){  
    Serial.println("turning left");  
    stepper.setSpeed(ROTATION_RPM);  
    stepper.step(-1 * STEPS);  
}  
delay(20);  
}
```