LAMPIRAN

INSTRUKSI DAN MANUAL BOOK

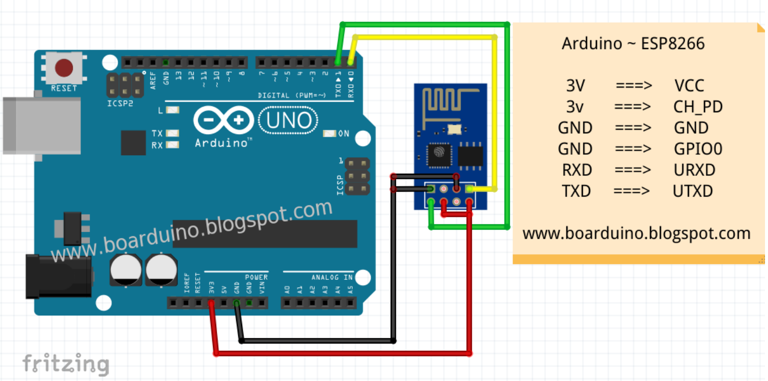
Pengisian flashing firmware ESP 8266-01

Langkah-langkahnya sebagai berikut :

1. Download terlebih dahulu ESPFlash tool dan firmware versi 0018000902-AI03
2. Lalu upload sketch Bare minimum dibawah ini kw Board Arduino

```
void setup() {
// Ini dibiarkan kosong .
}

void loop() {
  // Ini juga dikosongkan .
}
```

3. Susunlah rangkaian Seperti dibawah ini untul flashing firmware



Arduino ~ ESP8266

3V	===>	VCC
3v	===>	CH_PD
GND	===>	GND
GND	===>	GPIO0
RXD	===>	URXD
TXD	===>	UTXD

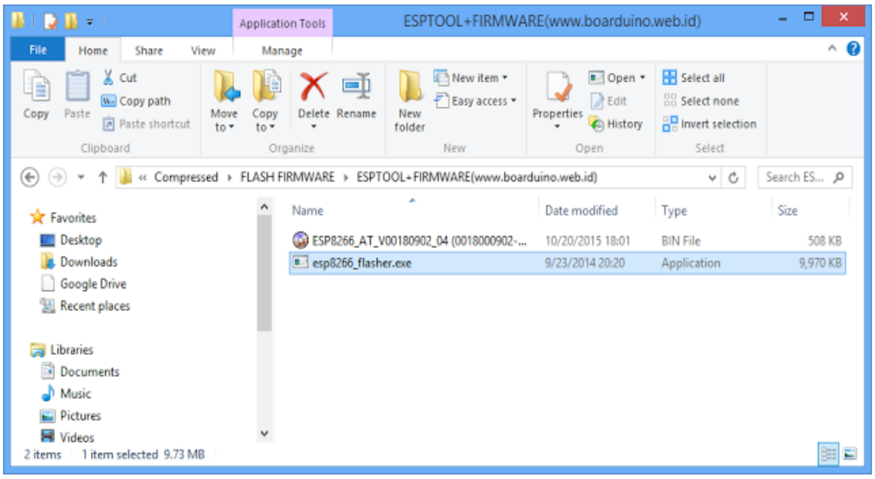
www.boarduino.blogspot.com

Gambar 3.4 Koneksi arduino dengan ESP 8266-01

Keterangan konfigurasi kabel :

6. Pin VCC pada Arduino dihubungkan ke pin VCC dan CH-PD Modul ESP 8266
7. Pin GND pada Arduino dihubungkan ke GND dan GPIO0 pada ESP 8266
8. Pin TXD pada Arduino dihubungkan ke UTXD pada modul ESP 8266
9. Pin RXD pada Arduino dihubungkan ke URXD pada modul ESP 8266

E Setelah itu extract file yang telah didownload sehingga muncul seperti gambar berikut



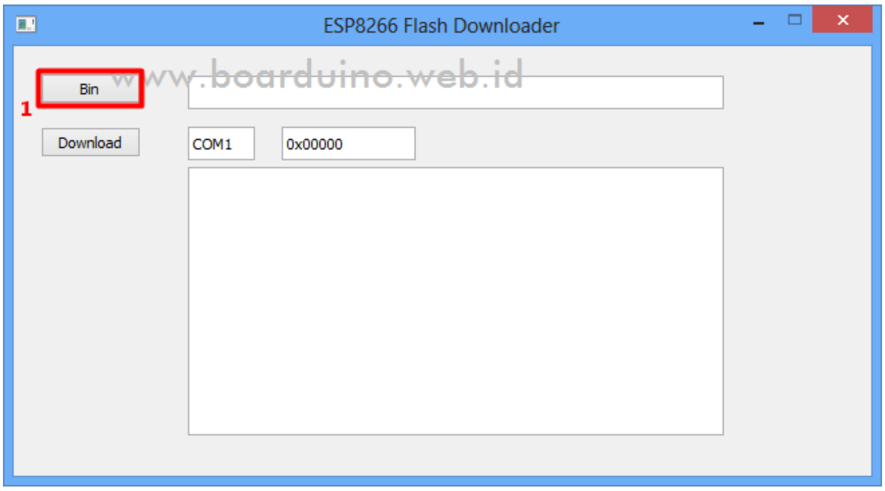
« Compressed › FLASH FIRMWARE › ESPTOOL+FIRMWARE(www.boarduino.web.id)

Name	Date modified	Type	Size
ESP8266_AT_V00180902_04 (0018000902-...	10/20/2015 18:01	BIN File	508 KB
esp8266_flasher.exe	9/23/2014 20:20	Application	9,970 KB

2 items   1 item selected 9.73 MB

Gambar 3.5 Folder Software ESP Flasher

F Setelah diextract lalu double klik pada ESP 8266_flasher.exe lalau klik button Bin.

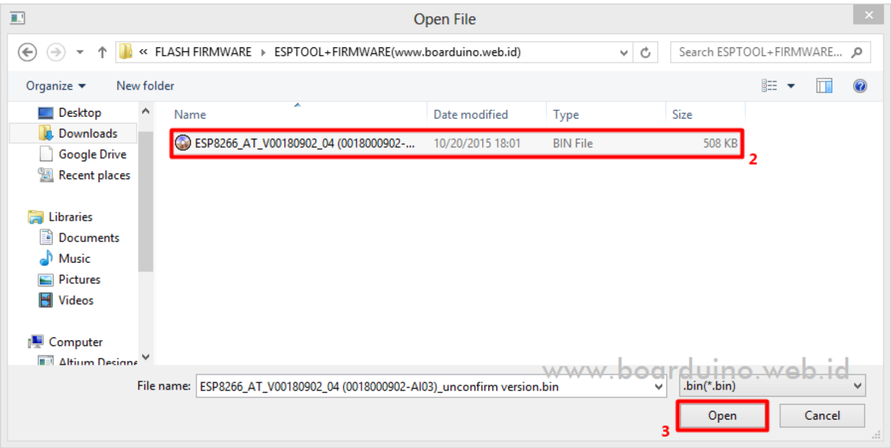


COM1

0x00000

Gambar 3.6 Software ESP 8266 Flash Downloader

G Selanjutnya pilih file firmware yang akan diflash, lalau klik open



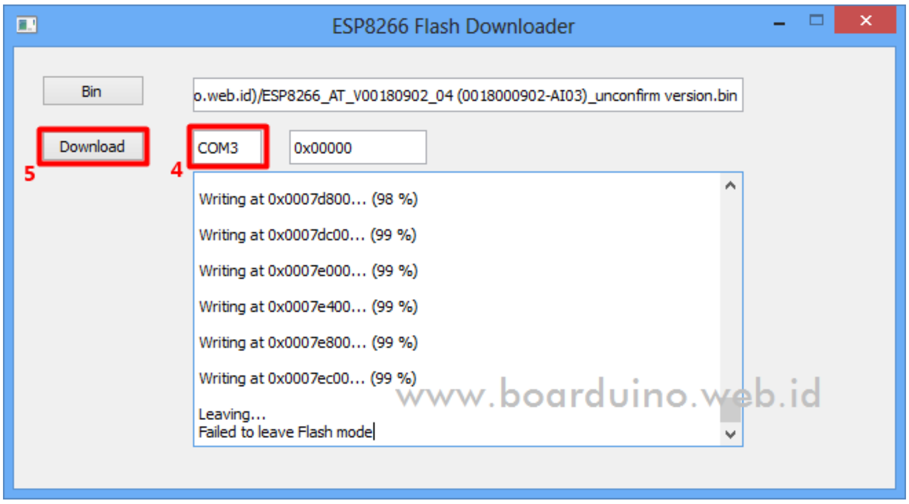
Name	Date modified	Type	Size
ESP8266_AT_V00180902_04 (0018000902-...	10/20/2015 18:01	BIN File	508 KB

File name: ESP8266_AT_V00180902_04 (0018000902-AI03)_unconfirm version.bin

.bin(*.bin)

Gambar 3.7 file firmware ESP 8266-01

H Lalu sesuaikan COM Port Arduino yang terdeteksi di computer, lalu klik download



o.web.id)/ESP8266_AT_V00180902_04 (0018000902-AI03)_unconfirm version.bin

COM3

0x00000

```
Writing at 0x0007d800... (98 %)
Writing at 0x0007dc00... (99 %)
Writing at 0x0007e000... (99 %)
Writing at 0x0007e400... (99 %)
Writing at 0x0007e800... (99 %)
Writing at 0x0007ec00... (99 %)
Leaving...
Failed to leave Flash mode
```

Gambar 3.8 Proses load firmware ESP 8266-01

I Setelah berhasil flashingnya, cabut kabel GPIO0 yang terhubung GND arduino

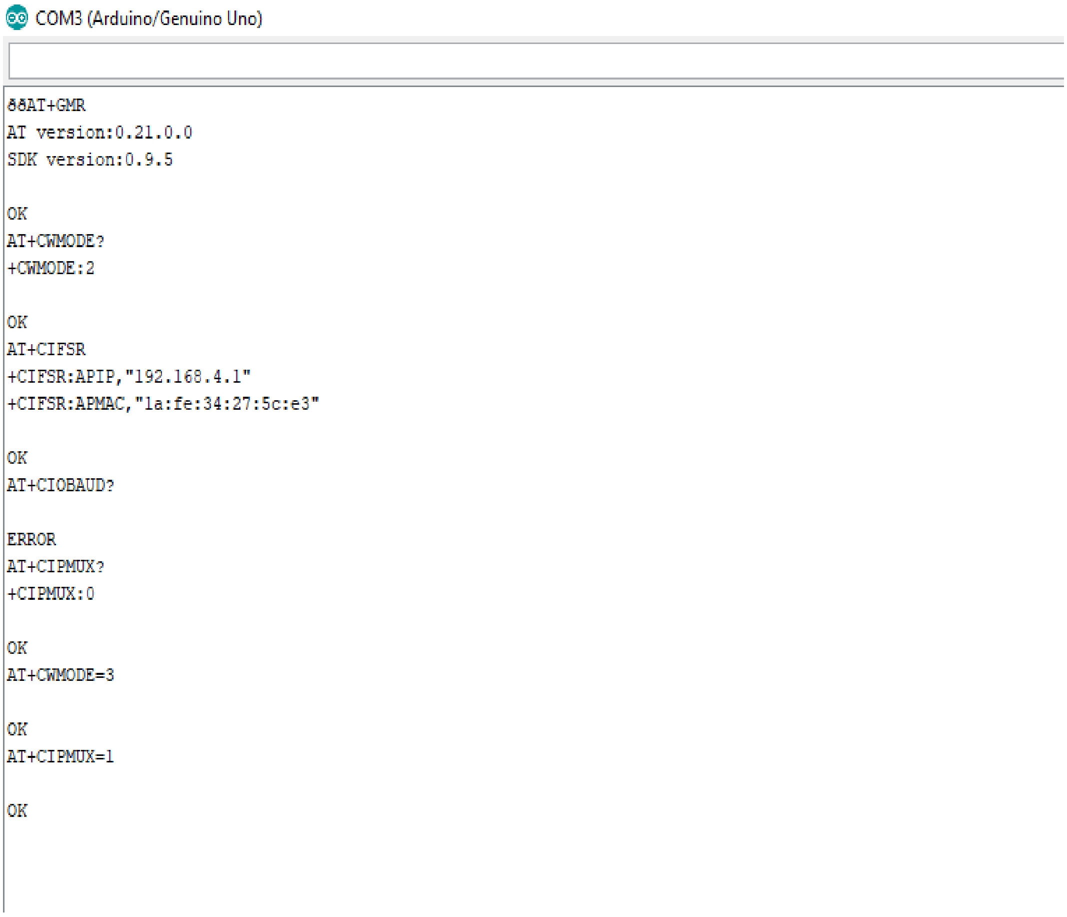
J Lalu buka serial monitor Arduino IDE dan set Both NL dan CR dengan baudrate 9600

K Lalu coba mengecek dengan mengirimkan perintah AT, AT+RST dan AT +GMR

Terhubung ke Jaringan WIFI

Setting At command yang wajib dilakukan supaya bisa terhubung sempurna dengan blynk server melalui wifi yaitu

1. AT +CWMODE=3 (Mode gabungan AP dan STA)
2. AT + CIOBAUD=9600 (mengeset baudrate menjadi 9600)
3. AT_CIPMUX=1 (Tipe TCP/UDP connections multiple)



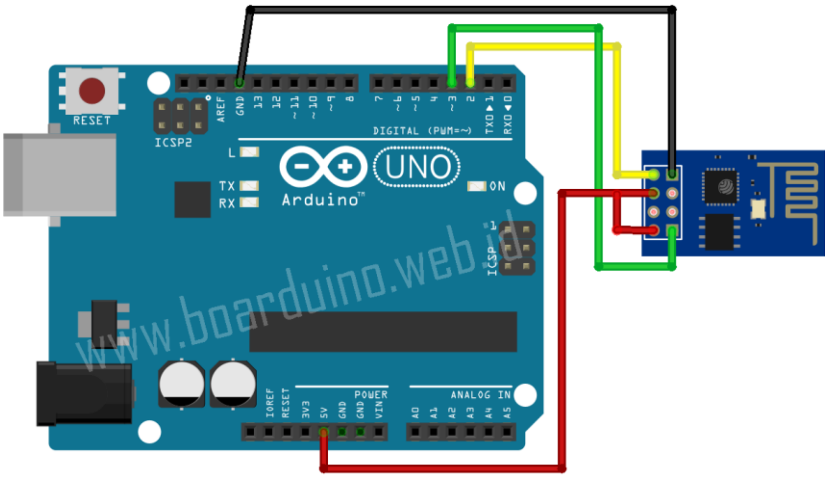
COM3 (Arduino/Genuino Uno)

```
88AT+GMR
AT version:0.21.0.0
SDK version:0.9.5

OK
AT+CWMODE?
+CWMODE:2

OK
AT+CIFSR
+CIFSR:APIP,"192.168.4.1"
+CIFSR:APMAC,"1a:fe:34:27:5c:e3"

OK
AT+CIOBAUD?

ERROR
AT+CIPMUX?
+CIPMUX:0

OK
AT+CWMODE=3

OK
AT+CIPMUX=1

OK
```

Gambar 3.9 setting mode ESP 8266-01 dengan AT Command

Kemudian tahap selanjutnya merubah rangkaian

1. Menyusun rangkaian seperti gambar berikut



Gambar 3.10 Konfigurasi arduino uno dngan ESP 8266-01

2. Setelah itu sambungkan arduino pada Laptop/PC dengan menggunakan kabel serial
3. Download dan Install program Arduino IDE di Arduino.cc
4. Jalankan Program Arduino
5. Klik Menu “tool→Board-→Arduino Uno”
6. Klik Menu “tool-→Port-→(Pilih Port arduino yang terdeteksi di computer)
7. Lalu masukkan Program kontrol arduino melalui Blynk

Tabel 3.2 Koneksi PIN arduino uno dengan ESP 8266-01

Arduino	**ESP 8266**
pin TX	pin RXD
pin RX	pin TXD
pin GND	pin GPIO 0
-	pin GPIO 2
pin 3.3V	pin RESET
-	pin CH_PD
pin GND	pin GND
pin 3.3V	pin Vcc

PROGRAM ARDUINO UNO

```
#define BLYNK_DEBUG // Optional, this enables lots of prints
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
#include <TimeLib.h>
#include <WidgetRTC.h>
#include <DHT.h>
#include <LiquidCrystal.h>
#define DHTPIN 3          // PIN SINYAL SENSOR DHT
#define DHTTYPE DHT11     // DHT 11
//#define DHTTYPE DHT22     // DHT 22
LiquidCrystal lcd(13, 8, 7, 6, 5, 4);
const int relay1 =  A0; humidifier
const int relay2 =  A1; kipas
const int relay3 =  A5; lampu biasa
int humLowTrigger;
int tempHighTrigger;
char auth[] = "c2d143cab18d4c98bbbfce665e9a2f9e";
char ssid[] = "kangmashadi";
char pass[] = "polytron";
#include <WidgetLED.h>
#define TestLED 9                 // JALUR KE RELAY LAMPU TIMER
char Date[16];
char Time[16];
long startsecondswd;            //
long stopsecondswd;             //
long nowseconds;                //
bool isFirstConnect = true;
```

```
int alldays;
#include <SoftwareSerial.h>
SoftwareSerial EspSerial(10, 11); // RX, TX
#include  <TimerOne.h>          //
volatile boolean zero_cross=0;  //
int AC_pin = 12;                //
int brightness = 128;           // Dimming level (0-128)  0 = on, 128 = 0ff
int freqStep = 75;    // frekuensi 65 hindari kedip
const byte degreeSymbol = B11011111; //Simbol Degree
int autodimmer;
#define ESP8266_BAUD 9600
//#define ESP8266_BAUD 115200
BlynkTimer timer;
WidgetRTC rtc;
ESP8266 wifi(&EspSerial);
DHT dht(DHTPIN, DHTTYPE);
void updateHum(int param);
void updateTemp(int param);
BLYNK_WRITE(V5) {
  updateHum(param.asInt());
}
BLYNK_WRITE(V6) {
  updateTemp(param.asInt());
}
void Readdata()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    lcd.print("ERROR");
    return;
```

```
}
  lcd.setCursor(0, 0); //
  lcd.print("TMP:");
  lcd.print((int)t);
  lcd.write(degreeSymbol); // Tulis simbol derajat
  lcd.print("C");
  //menampilkan nilai suhu pada LCD
  lcd.setCursor(9, 0);
  lcd.print("HUM:");
  lcd.print((int)h);
  lcd.println("%");
  lcd.setCursor(0, 1);
  lcd.print("JAM:");
  if(hour() < 10)
  {
  lcd.print("0");
  }
  lcd.print(hour(), DEC); //Print hour
  lcd.print(':');
  if(minute() < 10)
  {
  lcd.print("0");
  }
  lcd.print(minute(), DEC); //Print min
Blynk.virtualWrite(V15, h);
Blynk.virtualWrite(V16, t);
Blynk.virtualWrite(V25, humLowTrigger);
Blynk.virtualWrite(V24, tempHighTrigger);
if(h < humLowTrigger) {
    digitalWrite(relay1, LOW);
    Blynk.virtualWrite(V26, 255);
  } else {
      digitalWrite(relay1, HIGH);
      Blynk.virtualWrite(V26, 0);
  }
  if(t < tempHighTrigger) {
      digitalWrite(relay2, HIGH);
      Blynk.virtualWrite(V27, 0);
  } else {
      digitalWrite(relay2, LOW);
      Blynk.virtualWrite(V27, 255);
  }
}
void updateHum(int param) {
humLowTrigger = param;
}
void updateTemp(int param){
tempHighTrigger = param;
}
void zero_cross_detect() {
  zero_cross = true;
  i=0;
  digitalWrite(AC_pin, LOW);       // turn off TRIAC (and AC)
}
// Turn on the TRIAC at the appropriate time
void dim_check() {
  if(zero_cross == true) {
    if(i>=brightness) {
      digitalWrite(AC_pin, HIGH); // turn on light
      i=0;  // reset time step counter
      zero_cross = false; //reset zero cross detection
    }
  else {
      i++; // increment time step counter
    }
  }
}
void setup()
{
  lcd.begin(16, 2);
  dht.begin();
  pinMode(AC_pin, OUTPUT);                          // Set the Triac pin as output
  attachInterrupt(0, zero_cross_detect, RISING);    // Attach an Interupt to Pin 2 (interupt 0)
  Timer1.initialize(freqStep);                      //
  Timer1.attachInterrupt(dim_check, freqStep);
  Serial.begin(9600);
  EspSerial.begin(ESP8266_BAUD);
  delay(10);  // stabil
  Blynk.begin(auth, wifi, ssid, pass);
  //Blynk.syncAll();
  timer.setInterval(10000, Readdata);  // stabil MEMANGGIL VOID READ
  pinMode(relay1, OUTPUT);
  digitalWrite(relay1, HIGH);
  pinMode(relay2, OUTPUT);
  digitalWrite(relay2, HIGH);
  pinMode(relay3, OUTPUT);
  digitalWrite(relay3, HIGH);

  int humLowTrigger = 65;
  int TempHighTrigger = 20;
  pinMode(TestLED, OUTPUT);
  digitalWrite(TestLED, HIGH); // set RELAY LAMPU TIMER OFF
  int mytimeout = millis() / 1000;
  while (Blynk.connect() == false) { //
  if((millis() / 1000) > mytimeout + 8){ //
  break;
    }
  }
  rtc.begin();
  timer.setInterval(30000L, activetoday);  //  MEMANGGIL VOID AKTIVE
  timer.setInterval(20000L, reconnectBlynk);  //    MEMANGGIL RECONEK
}
BLYNK_CONNECTED() {
   Blynk.syncAll();
   if (isFirstConnect) {
   Blynk.notify("TIMER STARTING!!!!");
   rtc.begin();
    isFirstConnect = false;
  }
}
void activetoday(){        // check if schedule should run today
  if(year() != 1970){
   if (alldays==1) {
   Blynk.syncVirtual(V8); // sync timeinput widget

    }
  }
}

void reconnectBlynk() {
  if (!Blynk.connected()) {
    if(Blynk.connect()) {
     BLYNK_LOG("Reconnected");
    } else {
      BLYNK_LOG("Not reconnected");
    }
  }
}
BLYNK_WRITE(V3)//      slider
{
 int brillo = param.asInt();
 if (autodimmer == 0) //
 brightness=brillo;

}






BLYNK_WRITE(V7)  // Manual lampu BIASA
{
  if (param.asInt()==1) {
    digitalWrite(relay3, LOW);

  } else {
     digitalWrite(relay3, HIGH);

 }
}
BLYNK_WRITE(V2)  // LAMPU TIMER ON-OFF Manual
{
    if (param.asInt()==1) {
     digitalWrite(TestLED, LOW); // set LED ON
     lcd.setCursor(10, 1);
     lcd.print("L--ON");
  }
     if (param.asInt()==0)  {
     digitalWrite(TestLED, HIGH); // set LED OFF
     lcd.setCursor(10, 1);
     lcd.print("L-OFF");

     }
    }
BLYNK_WRITE(V9)  // SET TIMER ON-OFF
{
 if (param.asInt()==1) {
   alldays=1;
 }
 if (param.asInt()==0) {
  alldays=0;
 }
}
BLYNK_WRITE(V8)//All days
{
  if (alldays==1) {
  TimeInputParam t(param);
  int dayadjustment = -1;
  if(weekday() == 1){
   dayadjustment =  6; //
   }
  if(t.isWeekdaySelected(weekday() + dayadjustment)){ //
  for (int i = 1; i <= 7; i++) {  //
  if (t.isWeekdaySelected(i)) {
        }
      }
    nowseconds = ((hour() * 3600) + (minute() * 60) + second())
   // nowseconds = long(hour() * 3600L) + long(minute() * 60L) + long
(second());
    startsecondswd = (t.getStartHour() * 3600) + (t.getStartMinute() * 60);
    stopsecondswd = (t.getStopHour() * 3600) + (t.getStopMinute() * 60);
    //Serial.println(startsecondswd);  // used for debugging
    if(startsecondswd ==  stopsecondswd) {
        digitalWrite(TestLED, HIGH); // set LED OFF
        Blynk.virtualWrite(V2, 0);
        lcd.setCursor(10, 1);
        lcd.print("L-OFF");

    }
    if(startsecondswd < stopsecondswd){    //
         if ((nowseconds >= startsecondswd) && (nowseconds < stopsecondswd) ){
          digitalWrite(TestLED, LOW); // set LED On
          Blynk.virtualWrite(V2, 1);
          //Blynk.virtualWrite(V9, 1);
          lcd.setCursor(10, 1);
          lcd.print("L--ON");
         }
          else if (nowseconds >= stopsecondswd){
          digitalWrite(TestLED, HIGH); // set LED
          Blynk.virtualWrite(V2, 0);
          lcd.setCursor(10, 1);
          lcd.print("L-OFF");
          }
          else  {
          digitalWrite(TestLED, HIGH); // set LED OFF
```

```
     Blynk.virtualWrite(V2, 0);
      lcd.setCursor(10, 1);
      lcd.print("L-OFF");
      }
  }
  if(startsecondswd > stopsecondswd){    //
     if ((nowseconds >= startsecondswd) && (nowseconds <= 82800)){
      digitalWrite(TestLED, LOW); // set LED On
      Blynk.virtualWrite(V2, 1);
    //  Blynk.virtualWrite(V9, 1);
      lcd.setCursor(10, 1);
      lcd.print("L--ON");
     }
      else if (nowseconds < stopsecondswd){
      digitalWrite(TestLED, LOW); // set LED On
      Blynk.virtualWrite(V2, 1);
      // Blynk.virtualWrite(V9, 1);
      lcd.setCursor(10, 1);
      lcd.print("L--ON");
      }
      else if (nowseconds >= stopsecondswd && nowseconds < startsecondswd )
{
      digitalWrite(TestLED, HIGH); // set LED OFF
      Blynk.virtualWrite(V2, 0);
      lcd.setCursor(10, 1);
      lcd.print("L-OFF");



    }
  }
 }
```

```
  else{


    // nothing to do today, check again in 30 SECONDS time
  }
  //terminal.println();
 }
}

void loop()
{
  if (Blynk.connected()) {
    Blynk.run();
  }
  timer.run();
   yield();
}
```