

Interfacing of ESP8266 board with Blynk Console to control LEDs.

Steps for interfacing:

1. Programming on Blynk Console
 - a. Create new template
 - b. Name the template
 - c. Add the widget for LED -
2. Programming ESP8266 in Arduino IDE
3. Sharing credential of Blynk console in the ESP8266 program
4. Sharing Wifi credentials in the ESP8266 program.

Program:

```
// copy the three credentials given below from blynk console
#define BLYNK_TEMPLATE_ID "*****"
#define BLYNK_TEMPLATE_NAME "/////////"
#define BLYNK_AUTH_TOKEN "+++++++++++"
#define BLYNK_PRINT Serial

//hardWare Specifics
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
char ssid[] = "*****"; // "WIFI NAME";
char pass[] = "#####"; // "WIFI PASSWORD"; // Set password to "" for
open networks.

bool fetch_blynk_state = true; //true or false
#define wifiled LED_BUILTIN //D0int
int wifiFlag = 0;

// (NodeMcu OutPuts)
// define the GPIO connected with LED and Relays
#define RED_LED_Pin 16 //D0 esp8266's Board Pin diagram, Here GPIO Number is used
#define GREEN_LED_Pin 14 //D1 esp8266's Board Pin diagram, Here GPIO Number is used

// the virtual pins
#define VPIN_SWITCH_1 V0 // DataStream Virtual pin for Red LED's Widget Switch
#define VPIN_SWITCH_2 V1 // DataStream Virtual pin for Green LED's Widget Switch

bool RED_LED_State = LOW; //Define integer to remember the toggle state for RED LED
bool GREEN_LED_State = LOW; //Define integer to remember the toggle state for GREEN LED
BlynkTimer timer;
```

```

BLYNK_WRITE(VPIN_SWITCH_1) // Executes when the value of virtual pin 0 changes
{
    RED_LED_State = param.asInt();
    if( RED_LED_State == 1){ digitalWrite(RED_LED_Pin,HIGH); delay(10);}
    if( RED_LED_State == 0){ digitalWrite(RED_LED_Pin,LOW ); delay(10);}
    digitalWrite(RED_LED_Pin, RED_LED_State);
}

```

```

BLYNK_WRITE(VPIN_SWITCH_2) // Executes when the value of virtual pin 0 changes
{
    GREEN_LED_State = param.asInt();
    if( GREEN_LED_State == 1){ digitalWrite(GREEN_LED_Pin,HIGH); delay(10);}
    if( GREEN_LED_State == 0){ digitalWrite(GREEN_LED_Pin,LOW ); delay(10);}
    digitalWrite(GREEN_LED_Pin, GREEN_LED_State);
}

```

```

BLYNK_CONNECTED() {
    // Request the latest state from the server
    if (fetch_blynk_state){
        Blynk.syncVirtual(VPIN_SWITCH_1);
        Blynk.syncVirtual(VPIN_SWITCH_2);
    }
}

```

```

void checkBlynkStatus() { // called every 2 seconds by SimpleTimer

```

```

    bool isconnected = Blynk.connected();
    if (isconnected == false) {
        Serial.println("Blynk Not Connected");
        digitalWrite(wifiLed, HIGH);
    }
    if (isconnected == true) {
        if (!fetch_blynk_state){
            Blynk.virtualWrite(VPIN_SWITCH_1, RED_LED_State );
            Blynk.virtualWrite(VPIN_SWITCH_2, GREEN_LED_State);
        }

        wifiFlag = 0;
        digitalWrite(wifiLed, LOW);
        Serial.println("Blynk Connected");
    }
}

```

```

void setup()

```

```

{
    // Debug console
    Serial.begin(115200);
    pinMode(wifiLed, OUTPUT);
    pinMode(RED_LED_Pin, OUTPUT); // Initialise digital pin 5 as an output pin
    pinMode(GREEN_LED_Pin, OUTPUT); // Initialise digital pin 4 as an output pin

    //During Starting all LED are Turned OFF
    digitalWrite(RED_LED_Pin, LOW);
    digitalWrite(GREEN_LED_Pin, LOW);

    Blynk.begin(auth, ssid, pass);
    timer.setInterval(2000L, checkBlynkStatus); // check if Blynk server is
connected every 2 seconds
    delay(1000);

    if (!fetch_blynk_state){
        Blynk.virtualWrite(VPIN_SWITCH_1, RED_LED_State );
        Blynk.virtualWrite(VPIN_SWITCH_2, GREEN_LED_State);
    }
}

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

```

Programming of ESP8266 board to display output of DHT11 on Blyn1 console.
Steps for interfacing:

1. Programming on Blynk Console
 - a. Create new template
 - b. Name the template
 - c. Add the widget for LED -
2. Programming ESP8266 in Arduino IDE
3. Sharing credential of Blynk console in the ESP8266 program
4. Sharing Wifi credentials in the ESP8266 program.

```
/**
 * DHT11 Sensor Reader
 * This sketch reads temperature and humidity data from the DHT11 sensor and
 prints the values to the serial port.
 * It also handles potential error states that might occur during reading.
 *
 */

#define BLYNK_TEMPLATE_ID "*****"
#define BLYNK_TEMPLATE_NAME "+++++"
#define BLYNK_AUTH_TOKEN "scdgfdgfnmmbhmn"

// Include the DHT11 library for interfacing with the sensor.
#include <DHT11.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
char ssid[] = "+++++"; // "WIFI NAME";
char pass[] = "*****"; // "WIFI PASSWORD"; // Set password to ""
for open networks.
// Create an instance of the DHT11 class.
// - For ESP8266: Connect the sensor to GPIO2 or D4.
DHT11 dht11(2);

void setup() {
  // Initialize serial communication to allow debugging and data readout.
  // Using a baud rate of 9600 bps.
  Serial.begin(9600);
  WiFi.begin(ssid,pass);
  while(WiFi.status() != WL_CONNECTED){
    delay(1000);
    Serial.println("Connecting to WiFi");
  }
```

```

    }
    Blynk.begin(auth,ssid,pass);
}

void loop() {
    int temperature = 0;
    int humidity = 0;

    // read the temperature and humidity values from the DHT11 sensor.
    int result = dht11.readTemperatureHumidity(temperature, humidity);

    if (result == 0) {
        Serial.print("Temperature: ");
        Serial.print(temperature);
        Serial.print(" °C\tHumidity: ");
        Serial.print(humidity);
        Serial.println(" %");
        Blynk.virtualWrite(V0,temperature);
        Blynk.virtualWrite(V1,humidity);
    } else {
        // Print error message based on the error code.
        Serial.println(DHT11::getErrorString(result));
    }
}

```