Pseudo code:

Import lib and get blynk authtoken.

*import blynklib*

*BLYNK\_AUTH = ' ' #insert auth token*

*blynk = blynklib.Blynk(BLYNK\_AUTH)*

Wifi connection:

*import network*

*WIFI\_SSID = 'YourWifiSSID'  
WIFI\_PASS = 'YourWifiPassword'*

*wifi = network.WLAN(network.STA\_IF)  
wifi.active(True)  
wifi.connect(WIFI\_SSID, WIFI\_PASS)*

*# check if board connected   
connect\_status = wifi.isconnected()*

*From <*[*https://github.com/blynkkk/lib-python/tree/master/examples/esp32*](https://github.com/blynkkk/lib-python/tree/master/examples/esp32)*>*

Read sensor data as "temperature"

|  |  |
| --- | --- |
|  | *T\_VPIN = 3 # bind vpin 3 in blynk app* |
|  | *dht22 = dht.DHT22(Pin(4, Pin.IN, Pin.PULL\_UP)) #to be changed to our own sensor* |
|  | *@blynk.handle\_event('read V{}'.format(T\_VPIN))* |
|  | *def read\_handler(vpin):* |
|  | *temperature = 0.0* |
|  | *# read sensor data* |
|  | *try:* |
|  | *dht22.measure()* |
|  | *temperature = dht22.temperature()* |
|  | *except OSError as o\_err:* |
|  | *print("Unable to get DHT22 sensor data: '{}'".format(o\_err))* |
|  | *# change widget values and colors according read results* |
|  | *if temperature != 0.0 :* |
|  | *blynk.set\_property(T\_VPIN, 'color', T\_COLOR)* |
|  | *blynk.virtual\_write(T\_VPIN, temperature)* |
|  | *else:* |
|  | *# show widgets aka 'disabled' that mean we had errors during read sensor operation* |
|  | *blynk.set\_property(T\_VPIN, 'color', ERR\_COLOR)* |

*From <*[*https://github.com/blynkkk/lib-python/blob/master/examples/esp32/03\_temperature\_humidity\_dht22.py#L87*](https://github.com/blynkkk/lib-python/blob/master/examples/esp32/03_temperature_humidity_dht22.py#L87)*> altered*

Get user-defined default temperature as "thresholds"

*How to? By blynk app? That's to read vpin data?*

If temperature >= thresholds:

Send push notification to blynk app

Actions includes power up the fan or speed it up.

While true:

Blynk.run()