SCRIPT COMPLETI

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Networking;
using UnityEngine.UI;
using TMPro;


public class InviaDati : MonoBehaviour
{
    public string url;

    [SerializeField] private Slider _slider1;
    [SerializeField] private Slider _slider2;
    [SerializeField] private Slider _slider3;

    private string Valore1 = "0";
    private string Valore2 = "0";
    private string Valore3 = "0";

    private string Richiesta1;


    public void open()
    {

        //Debug.Log(Richiesta1);


        string Pacchetto1 = Valore1.PadLeft(3, '0'); //in que
        string Pacchetto2 = Valore2.PadLeft(3, '0');
        string Pacchetto3 = Valore3.PadLeft(3, '0');

        string Richiesta1 = url + Pacchetto1 + Pacchetto2 + P

        Debug.Log(Richiesta1);
        //Debug.Log(Valore2);
        StartCoroutine(GetRequest(Richiesta1)); //apro il lin

    }
    void Start()
    {
        _slider1.onValueChanged.AddListener((v) => {        /
            Valore1 = v.ToString("0");
            //Debug.Log(Valore1);
            open();
        });

        _slider2.onValueChanged.AddListener((v) =>
        {
            Valore2 = v.ToString("0");
            open();
            //Debug.Log(ValoreA);
        });
        _slider3.onValueChanged.AddListener((v) =>
        {
            Valore3 = v.ToString("0");
            open();
            //Debug.Log(ValoreA);
        });
    }

    IEnumerator GetRequest(string uri)
    {
        using (UnityWebRequest webRequest = UnityWebRequest.G
        {
            // Request and wait for the desired page.
            yield return webRequest.SendWebRequest();
        }
    }
}
```

```
/************************************************************

  You’ll need:
   - Blynk IoT app (download from App Store or Google Play)
   - ESP8266 board
   - Decide how to connect to Blynk
     (USB, Ethernet, Wi-Fi, Bluetooth, ...)

  There is a bunch of great example sketches included to show
  started. Think of them as LEGO bricks  and combine them as
  For example, take the Ethernet Shield sketch and combine it
  Servo example, or choose a USB sketch and add a code from S
  example.
 ************************************************************

// Template ID, Device Name and Auth Token are provided by th
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID           "TMPLQR6n6jpw"
#define BLYNK_DEVICE_NAME           "dispositivo"
#define BLYNK_AUTH_TOKEN            "ByGHH_9_m3LO7ecwVihSzNN6


// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>

Servo servo;
Servo servo1;
Servo servo2;
```

```
char auth[] = BLYNK_AUTH_TOKEN;


char ssid[] = "TIM-30140675";
char pass[] = "passWord Wifi";

int pinvalue;
int prova ;
bool acceso=true;

BLYNK_WRITE(V0){
  pinvalue=param.asInt();
  digitalWrite(LED_BUILTIN,pinvalue);
  Serial.print("\n");
  String Stringa = String(pinvalue);

  while (Stringa.length()!= 9)
  {
    Stringa= "0" + Stringa;
  }
  Serial.print("\n");
  Serial.print("x= " + Stringa.substring(0,3) + "  y= " + Str
  int valServo = Stringa.substring(0,3).toInt();
  servo.write(valServo);
  int valServo1 = Stringa.substring(3,6).toInt();
  servo1.write(valServo1);
  int valServo2 = Stringa.substring(6).toInt();
  servo1.write(valServo2);
}




void setup()
{
  // Debug console
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);     // Initialize the LED_BUI
  digitalWrite(LED_BUILTIN,1);
  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  digitalWrite(D0,HIGH);
  digitalWrite(D1,HIGH);
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8
  servo.attach(2);
  servo.write(0);
  servo1.attach(0);
  servo1.write(0);
  servo2.attach(1);
  servo2.write(1);
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sk
  // Check other examples on how to communicate with Blynk. R
  // to avoid delay() function!
}
```