



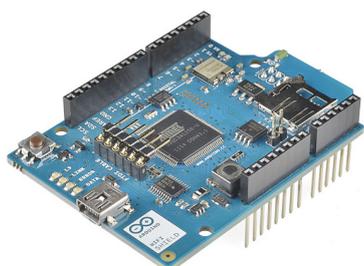
## Arduino

### Tutorial #14 - Sistema di Telepresenza

**Materiale utilizzato – Arduino UNO - WiFi Shield - motorino servo 180°- supporto per smartphone - smartphone.**

Per vedere tutti i tutorial: [www.marcopucci.it/arduino/](http://www.marcopucci.it/arduino/)

**N.B: [Clicca qui](#) per scaricare il file .zip con tutti gli sketch utilizzati per realizzare il sistema di telepresenza.**



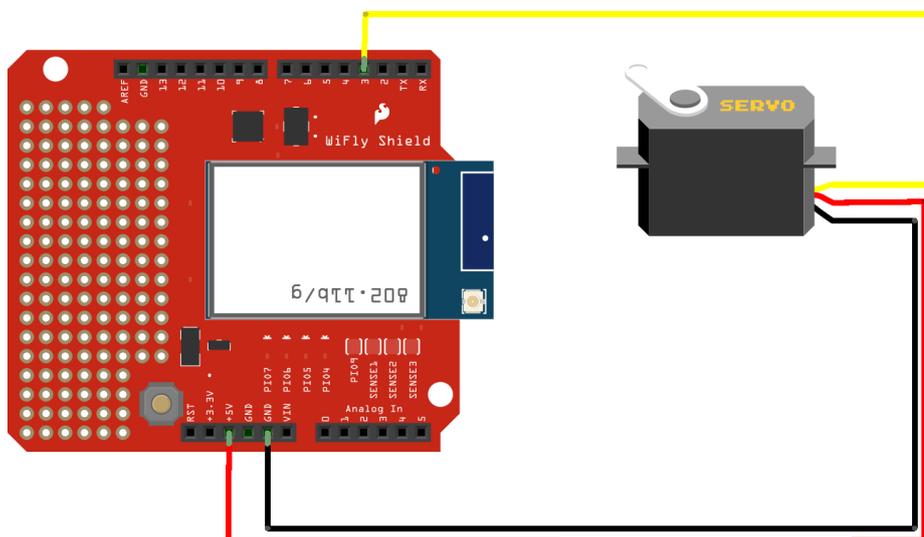
**N.B: Consiglio l'acquisto della versione ufficiale di Arduino della WiFi shield per problemi riscontrati con altre shield in commercio.**

In questo tutorial realizziamo un sistema per la Telepresenza. Attraverso una pagina web comandiamo un servo motore che fa ruotare un supporto per smartphone durante una videochiamata effettuata con Skype. In questo modo è possibile controllare cosa succede all'interno di una stanza, muovere la videocamera durante una videoconferenza o seguire cosa combina il gatto durante la nostra assenza. Se non avete familiarità con i motorini servo vi consiglio di leggere i precedenti tutorial - [clicca qui](#).

Il tutorial prevede una prima parte riguardante la connessione di Arduino con una pagina web tramite la WiFi Shield ufficiale ([clicca qui per acquistarla](#)). Attraverso questa pagina web possiamo gestire tre bottoni per comandare il movimento del supporto per smartphone che ruoterà in base al bottone premuto. La pagina web, in formato php, verrà creata in un server gratuito come [altestvista.org](http://altestvista.org). Quando clicchiamo su un bottone viene generato un file .txt con dei numeri. Arduino si conetterà a questa pagina e rileverà il valore scritto. In base al risultato il motorino cambierà posizione.

Se non avete familiarità con i file web non preoccupatevi perchè il codice verrà fornito con tutte le spiegazioni per cambiarlo a vostro piacimento.

### Schema elettrico



Inseriamo la WiFi Shield sopra la scheda Arduino e colleghiamo i tre cavetti seguendo lo schema a sinistra.



Scarichiamo dal sito di Arduino la libreria della WiFi Shield - oppure [clicca qui](#).  
Installiamo il file scaricato (scompattiamolo e copiamo il contenuto della cartella) nella directory **libraries** di Arduino.

Dato che è la prima volta che vediamo questo componente, vediamo qualche sketch all'interno degli esempi di Arduino.

Apriamo il software di Arduino. Click su File - Esempi - WiFi - **ScanNetworks**.

Questo file mostra le reti wireless presenti nel luogo in cui stiamo lavorando. Applodiamo il codice e apriamo il SerialMonitor. Se tutto funziona deve apparire una schermata simile all'immagine qui sotto.

```
COM44
Invia
MAC: 78:C4:E:2:35:97
Scanning available networks...
** Scan Networks **
number of available networks:7

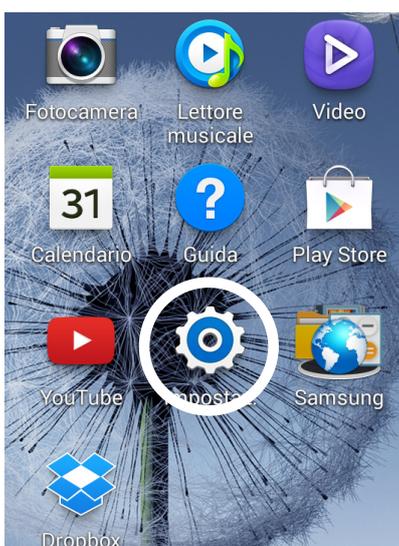
1) InfostradaWiFi-004153      Signal: -68 dBm Encryption: WPA2
2) FASTWEB-1-0021965D3528   Signal: -75 dBm Encryption: WPA2
3) Telecom-65572472         Signal: -78 dBm Encryption: WPA2
4) InfostradaWiFi-AD7448    Signal: -78 dBm Encryption: WPA2
5) Telecom-46133517        Signal: -83 dBm Encryption: WPA2
6) FASTWEB-1-PWmaMBn6ffsA   Signal: -84 dBm Encryption: WPA
```

Una lista delle reti wireless apparirà sul vostro monitor. Naturalmente per far funzionare la WiFi Shield abbiamo bisogno di una nostra rete wireless con user e password. Se in casa non avete una rete wireless potete generarne una dal vostro smartphone e collegarvi a questa.

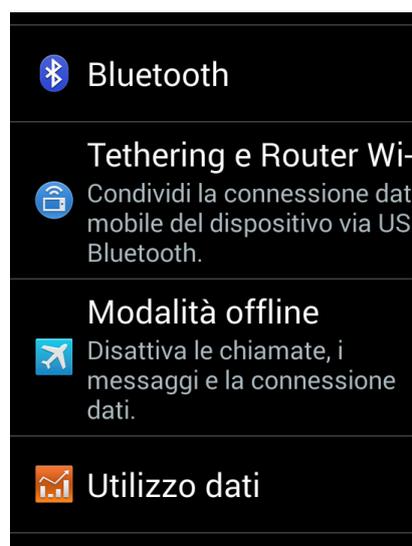
Vediamo molto velocemente come creare una rete HotSpot e come collegare la WiFi Shield ad essa (se invece avete una rete wireless potete saltare questo parte).

Il tutorial mostra la procedura per un Samsung S3. Tutti gli smartphone con connessione di dati attiva possono creare un HotSpot (se non siete in grado di farlo cercate in rete una guida per attivarlo).

## HotSpot



Click su Impostazioni



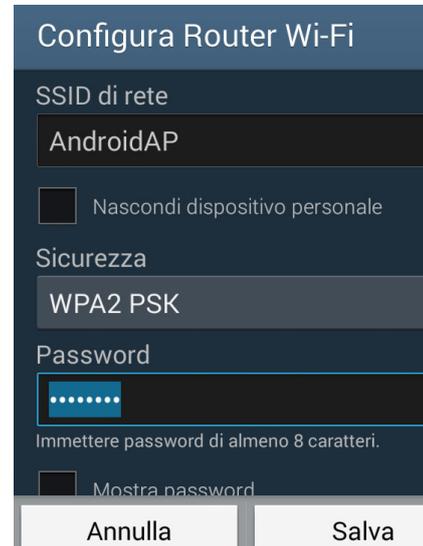
Tethering e Router Wi-Fi



Attivare Router Wi-Fi e cliccarci sopra per accedere alle proprietà.



Click su Configura



Inserire il nome della rete e la password. Salva

Apriamo un altro esempio di Arduino.

Click su File - Esempi - WiFi - **WifiWebClientRepeating**.

Questo file mostra come la shield WiFi riesce a collegarsi ad una rete wireless e leggere un file .txt presente su internet.

```
#include <SPI.h>
#include <WiFi.h>
```

```
char ssid[] = "yourNetwork";
char pass[] = "secretPassword";
```

Inseriamo il nome dell'HotSpot appena creato e la password oppure il nome della rete che abbiamo in casa.

Uplodiamo il codice e apriamo il SerialMonitor.

```
COM21
Attempting to connect to SSID: AndroidAP
SSID: AndroidAP
IP Address: 192.168.43.12
signal strength (RSSI):-69 dBm
connecting...
HTTP/1.1 200 OK
Server: nginx/1.4.2
Content-Type: text/plain
Last-Modified: Wed, 02 Oct 2013 13:46:55 GMT
ETag: "524c23cf-5"
Content-Length: 5
Accept-Ranges: bytes
Date: Wed, 03 Sep 2014 20:41:05 GMT
X-Varnish: 1252597555 1252595792
Age: 17
Via: 1.1 varnish
Connection: close
X-Cache: HIT

0105
```

Se sullo schermo appaiono delle scritte come queste a sinistra allora siamo pronti per continuare. Altrimenti controllate di avere inserito correttamente il nome della rete e la password oppure se il vostro smartphone ha un HotSpot attivo con connessione dati attiva.

Ogni 10 secondi Arduino si collega di nuovo al server e scarica i dati. Partiamo dall'analisi di questo sketch (realizzato da Tom Igoe) per poi semplificarlo e creare delle funzioni per collegarlo a un nostro sito per ricevere i comandi di gestione del motorino. Non verrà spiegato tutto il codice ma solamente i blocchi principali. Alcuni di essi verranno eliminati per semplificare lo sketch, altri verranno inseriti in alcune FUNZIONI in modo da essere richiamati in determinati momenti.

Se avete delle difficoltà con la spiegazione dei prossimi passaggi vi invito a continuare ugualmente.

L'importante è capire la logica delle varie funzioni e a mano a mano che il codice verrà semplificato apparirà tutto più chiaro.



```
#include <SPI.h>
#include <WiFi.h>
char ssid[] = "AndroidAP";
char pass[] = "xxx";
int keyIndex = 0;
int status = WL_IDLE_STATUS;

WiFiClient client;
char server[] = "www.arduino.cc";
unsigned long lastConnectionTime = 0;
boolean lastConnected = false;
const unsigned long postingInterval = 10*1000;

void setup() {
  Serial.begin(9600);
  while (!Serial) {
  }
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    while(true);
  }
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);
    delay(10000);
  }
  printWifiStatus();
}

void loop() {
  while (client.available()) {
    char c = client.read();
    Serial.write(c);
  }

  if (!client.connected() && lastConnected) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();
  }

  if(!client.connected() && (millis() - lastConnectionTime >
postingInterval)) {
    httpRequest();
  }
  lastConnected = client.connected();
}

void httpRequest() {
  if (client.connect(server, 80)) {
    Serial.println("connecting...");
    client.println("GET /latest.txt HTTP/1.1");
    client.println("Host: www.arduino.cc");
    client.println("User-Agent: arduino-ethernet");
    client.println("Connection: close");
    client.println();

    lastConnectionTime = millis();
  }
  else {
    Serial.println("connection failed");
    Serial.println("disconnecting.");
    client.stop();
  }
}

void printWifiStatus() {
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
}
```

Nome del sito a cui ci colleghiamo

Controllo se la Shield è presente

Eliminiamo: qui viene richiamata la funzione in fondo che stampa alcune delle informazioni apparse precedentemente.

Richiamo la funzione httpRequest()

Funzione fondamentale: quando viene chiamata verifica se il server è connesso ed esegue le operazioni successive. In questo caso apre il file latest.txt presente sul sito di Arduino.



## Aprirete il file **\_1\_WifiWebClientRepeating\_modificato**.

```
#include <TextFinder.h>
#include <SPI.h>
#include <WiFi.h>
char ssid[] = "xxx";
char pass[] = "xxx";
int keyIndex = 0;
int status = WL_IDLE_STATUS;

WiFiClient client;
TextFinder finder(client);
char server[] = "www.marcopucci.it";
unsigned long lastConnectionTime = 0;
boolean lastConnected = false;
const unsigned long postingInterval = 1*1000;

void setup() {
  Serial.begin(9600);
  while (!Serial) {
    ;
  }
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    while(true);
  }
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);
    delay(7000);
  }
}

void loop() {
  while (client.available()) {
    char c = client.read();
    Serial.write(c);
  }
  if (!client.connected() && lastConnected) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();
  }
  if(!client.connected() && (millis() -
lastConnectionTime > postingInterval)) {
    httpRequest();
    Serial.println("dato presente sul sito");
    finder.find("OK>");
    float value = finder.getFloat();
    Serial.println(value);
  }
  lastConnected = client.connected();
}

void httpRequest() {
  if (client.connect(server, 80)) {
    Serial.println("connecting...");
    client.println("GET /data.txt HTTP/1.1");
    client.println("Host: www.marcopucci.it");
    client.println("Connection: close");
    client.println();
    lastConnectionTime = millis();
  }
  else {
    Serial.println("connection failed");
    Serial.println("disconnecting.");
    client.stop();
  }
}
```

Importiamo la libreria per cercare un valore nel file .txt

Inseriamo i dati della rete wireless.

Nome del server a cui ci colleghiamo.  
Più avanti creiamo un vostro server.

Richiamo la funzione `httpRequest()` in basso. Dopo aver eseguito la funzione `httpRequest` il programma torna qui e cerca all'interno del file `data.txt` la parola `OK>`. Questo passaggio lo abbiamo analizzato nel tutorial n.9 - clicca qui se non lo hai letto. Il valore presente dopo la scritta `OK>` viene inserito nella variabile "value" che poi mandiamo in stampa.

Connessione al server e al file presente nella directory `www.marcopucci.it/data.txt`

Che cosa abbiamo fatto? Uplodiamo e apriamo il SerialMonitor di Arduino.



```
COM44
Attempting to connect to SSID:
connecting...
dato presente sul sito
2.00
```

Ci siamo collegati al sito [www.marcopucci.it](http://www.marcopucci.it) dove è presente un file chiamato `data.txt`

Se provate a inserire nel browser l'indirizzo [www.marcopucci.it/data.txt](http://www.marcopucci.it/data.txt) apparirà una scritta `OK>` seguita da un numero.

Arduino si collega a questo file ed esegue una ricerca per individuare la parola `"OK>"`. Il valore subito dopo questo testo viene inserito in una variabile che poi mandiamo in stampa.

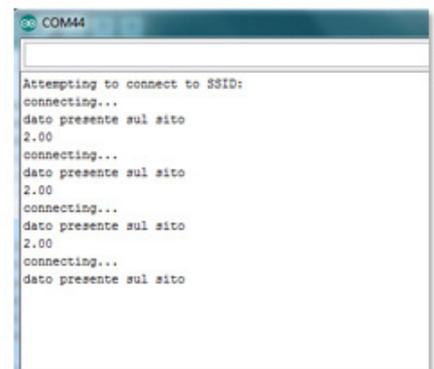
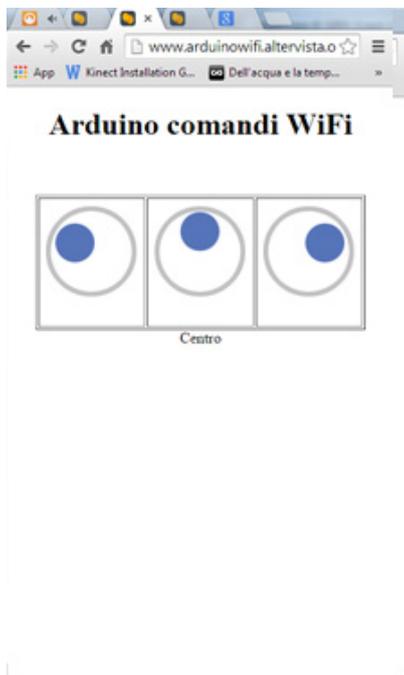
A sinistra l'immagine del SerialMonitor con presente il valore individuato.

Cos'è questo valore? Questo numero viene generato da un file `.php` con dei bottoni. Apriamo un'altra finestra del browser a questo indirizzo: <http://www.marcopucci.it/test.php>

Se premete il primo bottone e tornate nella pagina `marcopucci.it/data.txt` e aggiornate apparirà il numero 1. Premendo gli altri bottoni in valore cambierà nuovamente. In questo modo, attraverso una pagina web con dei pulsanti, riusciamo a scrivere su un file `.txt` che viene letto da Arduino. Nel prossimo sketch questi valori verranno analizzati e, in base al numero rilevato, faremo muovere un motorino.

Questo sistema di trasferimento dati ci permetterà di sviluppare numerose applicazioni dove Arduino è gestito tramite un file presente nel web e quindi da qualsiasi parte nel mondo.

Qui sotto uno schema riassuntivo del sistema di comunicazione.



Quando premo un bottone presente nella pagina `www.marcopucci.it/test8.php` (fig.1) viene modificato un file su `www.marcopucci.it/data.txt` (fig.2).

Arduino si collega a internet e individua il valore stampandolo sul proprio Serial Monitor (fig.3)



## Aprire il file **\_2\_WifiWebClientRepeating\_modificato\_servomotori**.

In questo sketch non facciamo altro che analizzare il valore della variabile "value". Attraverso tre controlli **IF** vediamo se il valore è uguale a "1", "2" o "3" (questi valori cambiano in base ai bottoni premuti nel file online test.php).

Utilizziamo tre funzioni che chiamiamo destra(), sinistra() e centro() per spostare il servo motore.

Se la variabile è uguale a 1 attiviamo la funzione sinistra() se il valore è uguale a 2 chiamiamo la funzione centro() altrimenti se è uguale a 3 la funzione destra().

Qui sotto la parte di codice all'interno della funzione loop() e le tre nuove funzioni, il resto è identico allo sketch precedente. Aggiungiamo la libreria servo e impostiamo il motorino servo sul pin 3.

Prima di Uplodarlo ricordatevi di inserire in alto il nome e la password della vostra connessione wireless.

```
void loop() {
  while (client.available()) {
    char c = client.read();
    Serial.write(c);
  }
  if (!client.connected() && lastConnected) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();
  }
  if (!client.connected() && (millis() - lastConnectionTime > postingInterval)) {
    httpRequest();
    Serial.println("dato presente sul sito");
    finder.find("OK>");
    float value = finder.getFloat();
    Serial.println(value);
    if (value==1){
      sinistra();
    }
    else if (value==2){
      centro();
    }
    else if (value==3){
      destra();
    }
  }
  lastConnected = client.connected();
}

void httpRequest() {
  ...
}

void centro(){
myservo1.write(90);
delay(2000);
}
void sinistra(){
myservo1.write(50);
delay(2000);
}
void destra(){
myservo1.write(130);
delay(2000);
}
```

[Richiamo la funzione per connettermi al sito](#)

Cerco il valore che appare nel file data.txt subito dopo la scritta "OK>"

Se il valore è uguale a 1 allora attivo la funzione sinistra() - vedere in fondo al codice

Se il valore è uguale a 2 allora attivo la funzione centro() - vedere in fondo al codice

Se il valore è uguale a 3 allora attivo la funzione destra() - vedere in fondo al codice

[Funzione per connetterci al sito. Per motivi di spazio non viene riportata ma è identica a quella dello sketch precedente](#)

Funzione centro() - muove il servo all'angolo 90 e si ferma

Funzione sinistra() - muove il servo all'angolo 50 e si ferma

Funzione destra() - muove il servo all'angolo 130 e si ferma

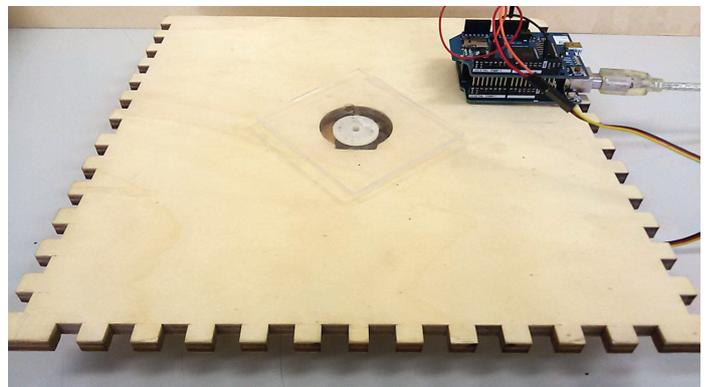
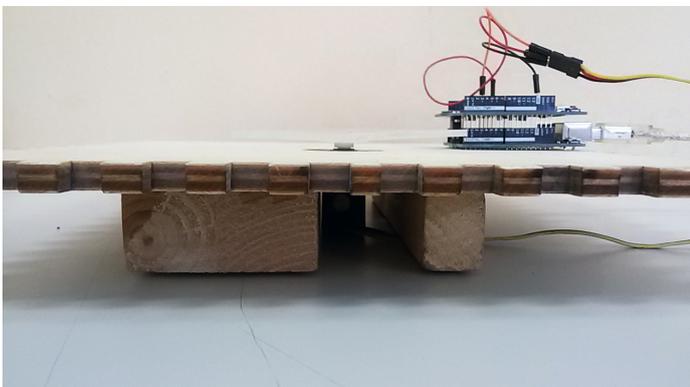


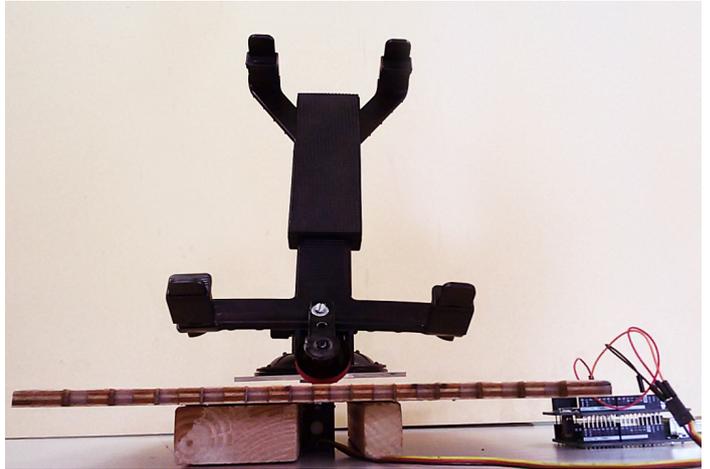
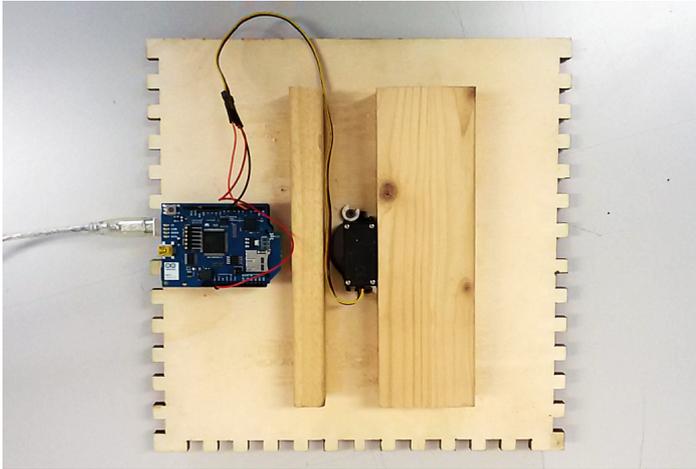
Ora che siamo in grado di gestire il movimento di un motorino servo tramite un sito web possiamo decidere di attaccare a questo motore qualsiasi oggetto. In questo caso, per completare il tutorial, creiamo una postazione di telepresenza ovvero muoviamo uno smartphone collegato in videochiamata Skype in modo da gestire l'angolazione della camera e inquadrare tutta la stanza a nostro piacere.

Per collegare lo smartphone al motorino dobbiamo acquistare un supporto per smartphone/tablet in vendita in qualsiasi negozio cinese o di elettronica come MediaWorld.

Qui sotto una piccola galleria per mostrarvi come ho realizzato la struttura mobile.

Nella prossima pagina vediamo invece come creare un vostro file .php e .txt e metterlo in rete in un server gratuito come altervista.org (se avete già un sito web potete caricare i due file presenti nella cartella direttamente nel vostro spazio web).







## Altervista.org

Altervista è un sito dove possiamo creare un nostro spazio web gratuitamente. Registrandosi viene fornito un dominio. Nel nostro caso utilizzeremo questo spazio per uplodare i due file data.txt e test.php.

Collegatevi al sito [www.altervista.org](http://www.altervista.org)

Inserite il nome del vostro sito nella casella di testo. Nel mio caso ho scelto come nome **arduinowifi.altervista.org**. Click su CONTINUA.

Nella pagina successiva riempite il form con i vostri contatti.  
Una volta ricevuta l'email di conferma, cliccate sul link per attivare il sito.

Torniamo al sito e clicchiamo su **PANNELLO** (in alto a destra). Inseriamo la user e la password ricevute per email.

Nella nuova schermata fate click su **GESTIONE FILE - VAI AI FILE**.

Click su **INVIA FILE**.

Click su **SELEZIONA FILE**. Selezioniamo nella nuova finestra il file data.txt e poi click su **INVIA FILE**.

La stessa operazione dobbiamo ripeterla per il file test.php e per le immagini presenti nella cartella che avete scaricato all'inizio del tutorial.

Quando avete finito, fate click su **TORNA ALLA GESTIONE FILE** (in basso). Qui sono presenti tutti i file caricati sul vostro sito.

Verifichiamo che tutto sia andato bene.

Apriamo il browser e inseriamo il link: <http://www.vostrosito.altervista.org/test.php>

Nel mio caso il link è: <http://www.arduinowifi.altervista.org/test.php>

Apparirà la pagina che abbiamo visto nel tutorial, solamente che ora è una pagina presente sul vostro sito e potete modificarla a vostro piacere.

Aprirete una seconda finestra nel browser e inserite il link:

<http://www.vostrosito.altervista.org/data.txt>

Anche in questo caso appare la pagina .txt che abbiamo visto in precedenza con la scritta OK> seguita da un numero.

Provate a cliccare i bottoni nella pagina test.php e vedete se il numero presente nel file <http://www.vostrosito.altervista.org/data.txt> cambia (ogni volta dovete aggiornare la pagina).

Se siete riusciti a completare questi passaggi, siete pronti per collegare Arduino al vostro sito.

Aprirete lo sketch **\_2\_WifiWebClientRepeating\_modificato\_servomotori** e modificate il nome del server.

```
char server[] = "www.marcopucci.it";
```

```
...
```

```
void httpRequest() {  
    if (client.connect(server, 80)) {  
        Serial.println("connecting...");  
        client.println("GET /data.txt HTTP/1.1");  
        client.println("Host: www.marcopucci.it");
```

All'inizio del codice cambiate il nome del mio sito con il vostro su altervista  
["www.vostrosito.altervista.org"](http://www.vostrosito.altervista.org)

Il nome va cambiato anche all'interno della funzione httpRequest()  
["www.vostrosito.altervista.org"](http://www.vostrosito.altervista.org)



Uplodate il nuovo codice.

In questo modo potete gestire qualsiasi componente elettronico a distanza.

Se realizzate qualche progetto, inviatemi le immagini o i video dei vostri lavori al seguente indirizzo: [puccimarco76@yahoo.it](mailto:puccimarco76@yahoo.it). I più interessanti verranno pubblicati sul sito e sulla pagina Facebook "Tutorial Arduino".

Terminiamo il tutorial con una veloce spiegazione del file test.php in modo da poterlo modificare a vostro piacere.

Aprirete il file **test.php** - Per aprirlo potete utilizzare il Blocco Note o qualsiasi altro editor di testo.

```
<!DOCTYPE html>
<html>
<head>
<title>Arduino WiFi</title>
</head>
<body>
<center>
<h1>Arduino comandi Wifi</h1>
<p><br/>

<table width="200" border="1">
  <tr>
    <td> <?php
      echo "<form action='#' method='post'>
        <input type='image' name='button2'?
src='sinistra.gif' value='Sinistra' />
        </form><br/>" ;?>
    </td>

<td><?php
      echo "<form action='#' method='post'>
        <input type='image' name='button1'?
src='centro.gif' value='Centro' />
        </form><br/>" ;?>
    </td>

<td><?php
      echo "<form action='#' method='post'>
        <input type='image' name='button3'?
src='destra.gif' value='Destra' />
        </form><br/>" ;?>
    </td>

  </tr>
</table>
```

La prima parte di codice si riferisce alla parte grafica in html

Scritta che appare al centro pagina.

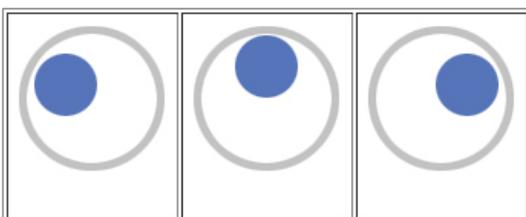
Inizia la tabella composta da tre colonne.

Nella prima colonna carico l'immagine sinistra.gif  
Se clicco sull'immagine assegno un valore al bottone =  
button2

Immagine centro.gif con bottone=button1

Immagine destra.gif con bottone=button3

## Arduino comandi WiFi



Centro

Nella figura la pagina creata con il codice html



```
<?php
if (isset($_POST['button1']))
{
    echo "Sinistra";
    $myFile = "data.txt";
    $fh = fopen($myFile, 'w');
    fwrite($fh, "OK>1<");
    fclose($fh);
}
if (isset($_POST['button2']))
{
    echo "Centro";
    $myFile = "data.txt";
    $fh = fopen($myFile, 'w');
    fwrite($fh, "OK>2<");
    fclose($fh);
}
if (isset($_POST['button3']))
{
    echo "Destra<br/>";
    $myFile = "data.txt";
    $fh = fopen($myFile, 'w');
    fwrite($fh, "OK>3<");
    fclose($fh);
}

?>
</p>
</center>
</body>
</html>
```

Vediamo ora la parte di codice php

Se ricevo un valore dal bottone uguale a "button1" scrivo sul file data.txt la scritta "OK>1"

Se ricevo un valore dal bottone uguale a "button2" scrivo sul file data.txt la scritta "OK>2"

Se ricevo un valore dal bottone uguale a "button3" scrivo sul file data.txt la scritta "OK>3"

Potete modificare il file a vostro piacere.

Se cambiate le immagini ricordatevi di uplodarle sul sito di altervista e cambiare il nome del file all'interno del codice.