

COMUNICAZIONE ANDROID ARDUINO

Prof. Fischetti Pietro

Per realizzare una semplice comunicazione tra Android e ArduinoUNO senza shield aggiuntive, si possono utilizzare i programmi (in C) per Windows:

wsocksvr.exe: un programma in C che crea un socket server in ascolto per default sulla porta 27000.

arduSer.exe: legge dati da tastiera e li trasmette via USB/Seriale a Arduino



Il cellulare viene configurato come hotspot e si connette il WiFi del PC al cellulare.

Sul PC leggiamo l'indirizzo IP della scheda WiFi tramite il comando ipconfig:

C:\ Prompt dei comandi

```
C:\TEMP>ipconfig

Configurazione IP di Windows

Scheda Ethernet Ethernet:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

Scheda LAN wireless Connessione alla rete locale (LAN)* 1:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::7daa:4a0c:4b6b:d156%22
    Indirizzo IPv4. . . . . : 192.168.137.1
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . :

Scheda LAN wireless Connessione alla rete locale (LAN)* 4:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

Scheda LAN wireless Wi-Fi:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::d0dc:ad79:5c3d:4d75%7
    Indirizzo IPv4. . . . . : 192.168.43.197
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.43.1

Scheda Ethernet Connessione di rete Bluetooth 2:
```

Sul PC dalla shell DOS si avviano i programmi con il seguente comando (controllare la porta COM su cui lavora Arduino, qui ad esempio e' la COM9):

```
C:\ Prompt dei comandi

C:\TEMP>wsocksrv | arduser -pCOM9 -mT
```

```
C:\ Prompt dei comandi

C:\TEMP>wsocksrv | arduser -pCOM9 -mT
Listen on Port:27000
Wait for connection...
[TX]BaudRate:9600 ByteSize:8 StopBits:0 Parity:0 DTR Ctrl:1.
#
```

Notare come si sono collegati i due programmi tramite la pipe (|). Questo modo molto comodo e' uno standard sia in Windows che in Linux e permette a programmi indipendenti di comunicare tra loro. Cioe' ad esempio se avessi tre programmi qualsiasi:

C:\>PROG1 | PROG2 | PROG3

L'uscita di PROG1 viene inviata dal Sistema Operativo all'input di PROG2 e l'uscita di questo diventa l'input di PROG3... etc.

A questo punto il PC e' pronto a ricevere i dati da Android. Verifichiamolo aprendo un'altra finestra DOS:

C:\ Prompt dei comandi

```
C:\TEMP>netstat -nao
```

Connessioni attive

Proto	Indirizzo locale	Indirizzo esterno	Stato	PID
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	1084
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:1801	0.0.0.0:0	LISTENING	4568
TCP	0.0.0.0:2103	0.0.0.0:0	LISTENING	4568
TCP	0.0.0.0:2105	0.0.0.0:0	LISTENING	4568
TCP	0.0.0.0:2107	0.0.0.0:0	LISTENING	4568
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING	3488
TCP	0.0.0.0:5357	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:7250	0.0.0.0:0	LISTENING	8400
TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING	1848
TCP	0.0.0.0:27000	0.0.0.0:0	LISTENING	15248
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING	936
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING	784
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING	2188
TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING	1820
TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING	2804
TCP	0.0.0.0:49669	0.0.0.0:0	LISTENING	3680
TCP	0.0.0.0:49672	0.0.0.0:0	LISTENING	4568
TCP	0.0.0.0:49679	0.0.0.0:0	LISTENING	916
TCP	127.0.0.1:5037	0.0.0.0:0	LISTENING	16228
TCP	127.0.0.1:5037	127.0.0.1:63772	ESTABLISHED	16228

Sì, il PC ha la porta 27000 aperta dal processo con PID 15248. Per conoscere il nome di questo PID aprire il programma Gestione Attività:

Gestione attività

File Opzioni Visualizza

Processi Prestazioni Cronologia applicazioni Avvio Utenti **Dettagli** Servizi

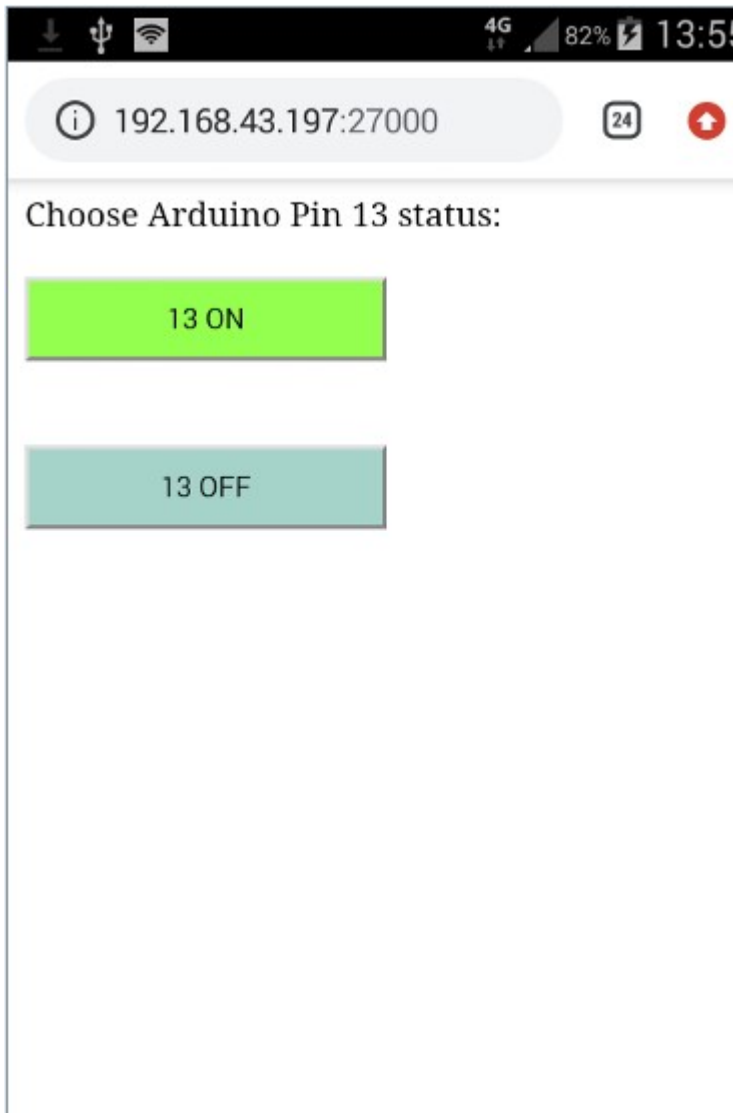
Nome	PID	Stato	Nome ute...	CPU	M
chrome.exe	13668	In esecuzione	acer	00	
chrome.exe	13868	In esecuzione	acer	00	
chrome.exe	14084	In esecuzione	acer	00	
chrome.exe	14116	In esecuzione	acer	00	
FoxitReader.exe	14176	In esecuzione	acer	00	
conhost.exe	14784	In esecuzione	acer	00	
chrome.exe	14996	In esecuzione	acer	00	
LockApp.exe	15108	Sospeso	acer	00	
wssocksv.exe	15248	In esecuzione	acer	00	
chrome.exe	15700	In esecuzione	acer	00	
conhost.exe	15924	In esecuzione	acer	00	
Taskmgr.exe	16004	In esecuzione	acer	02	
chrome.exe	16108	In esecuzione	acer	00	
cmd.exe	16132	In esecuzione	acer	00	
adb.exe	16228	In esecuzione	acer	11	
svchost.exe	16640	In esecuzione	SYSTEM	00	
chrome.exe	16836	In esecuzione	acer	00	
SearchUI.exe	16988	Sospeso	acer	00	
chrome.exe	17032	In esecuzione	acer	00	
javaw.exe	17048	In esecuzione	acer	04	
mspaint.exe	17784	In esecuzione	acer	00	
chrome.exe	17956	In esecuzione	acer	00	

Meno dettagli Termina attività

Il PID 15248 corrisponde proprio al programma wssocksv.exe.

Spostiamoci sul cellulare Android.

Se l'indirizzo IP del PC (fornito dall'Hotspot Android) sulla scheda WiFi e' ad esempio: 192.168.42.197 sul browser del cellulare si dovra' impostare l'URL: <http://192.168.43.198:27000>.



Una volta che viene inviata una richiesta di connessione dal browser del cellulare Android, all'IP e porta sulla quale è in ascolto wsocksrv.exe questo invia al client una pagina html (index.html personalizzabile) che contiene semplicemente 2 bottoni per l'accensione e lo spegnimento del led sul Pin 1 d Arduino.

```
Index.html
<!DOCTYPE html>
<html lang="en-US">
<head>
<title>Arduino PIN 13</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<form method="POST">
  Choose Arduino Pin 13 status:
  <button name="arg" type="submit" value="PIN13_1" style="padding: 10px 24px;width: 50%;display:
  block;margin-top: 20px;margin-bottom: 40px;background-color: #95FF4F;">13 ON</button>
  <button name="arg" type="submit" value="PIN13_0" style="padding: 10px 24px;width: 50%;display:
  block;background-color: #A5D3CA">13 OFF</button>
</form>
```

```
</body>
</html>
```

Se non si possiede un PC con connessione WiFi, ad esempio un Desktop, si può sempre utilizzare il browser sullo stesso PC e impostare l'URL all'indirizzo IP della scheda di rete.

Di seguito lo sketch Arduino per leggere comandi (PIN13_1/PIN13_0) dalla seriale e accendere/spegnere il Led su pin 13.

```
RXBlink.ino
#define BAUD 9600
#define led 13

String receivedString;

void setup() {
  Serial.begin(BAUD);
  pinMode(led, OUTPUT);
}

void loop() {
  if (Serial.available() > 0){
    receivedString = Serial.readStringUntil('\n');
    receivedString.trim();
    if (receivedString=="PIN13_1")
      digitalWrite(led, HIGH);
    else if (receivedString.equals("PIN13_0"))
      digitalWrite(led, LOW);
  }
}
```

Notifiche Arduino -> Android Tramite Cloud

Per poter inviare notifiche da Arduino (collegato al PC) su un telefonino Android si può utilizzare la piattaforma Cloud gratuita Firebase di Google. Dopo essersi registrati sul sito [Firebase.google.it](https://firebase.google.it) ci viene assegnata una chiave chiamata "Server Key". Poi occorre un'App Android che dovrà essere registrata su server Cloud. Sul nostro PC occorre un'applicazione che possa chiedere via Web al Cloud di inviare una notifica che specifichiamo, A questo scopo si può utilizzare l'applicazione gratuita per linea di comando curl (<https://curl.haxx.se/>). Se si vuole che il telefonino Android ricevi la notifica occorre che ci sia l'App suddetta installata (e registrata sul Cloud Firebase con nome pacchetto completo:it.mydev.myfcm), inoltre l'App come prima operazione dovrà richiedere al server Cloud un'ulteriore chiave (Token Key) che dovrà essere inviata in qualche modo all'applicazione presente sul PC(curl in questo caso) Nell'esempio seguente il programma arduer si pone in ascolto dei dati ricevuti da Arduino per poi spedirli tramite uno script utilizzando il tool curl.



4G



100%



10:18 PM

Send mail...



Aggiungi a
Dropbox



Bluetooth



Copia negli
appunti



E-mail



Gmail



Google+



Hangouts



Messaggi



Nel segnaposto <Server Key> bisogna sostituire il valore Fornito dalla Piattaforma Cloud, e nel segnaposto <App Key> il Token Fornito all'App dalla piattaforma Cloud.

```
C:/> arduser -pCOM9 -mR | fcm.bat
```