Pilotare le uscite GPIO di Raspberry tramite Server TCP/IP sviluppato in Java e client Android

Obiettivo: Accendere e spegnere tre LED tramite Raspberry via Server Java TCP/IP e Client Android.

Pilotare le uscite GPIO di Raspberry tramite Server TCP/IP sviluppato in Java e client Android

Componenti:

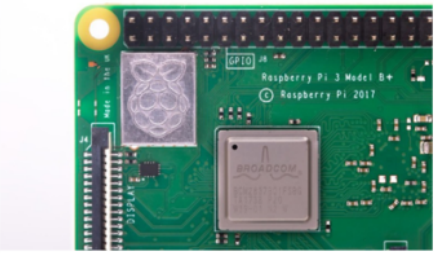
- Raspberry Pi 3 Model B+ oppure Raspberry Pi 4 Model B+
- N.1 LED Rosso
- N.1 LED Giallo
- N.1 LED Verde
- N.3 resistenze da 220 ohm

Teoria:

Alla base di questa esercitazione c'è Raspberry e la libreria Pi4J.

Raspberry Pi 3 Model B+ è la versione finale della famiglia Raspberry Pi 3 (https://www.raspberrypi.org/products/raspberry-pi-3-model-b-p

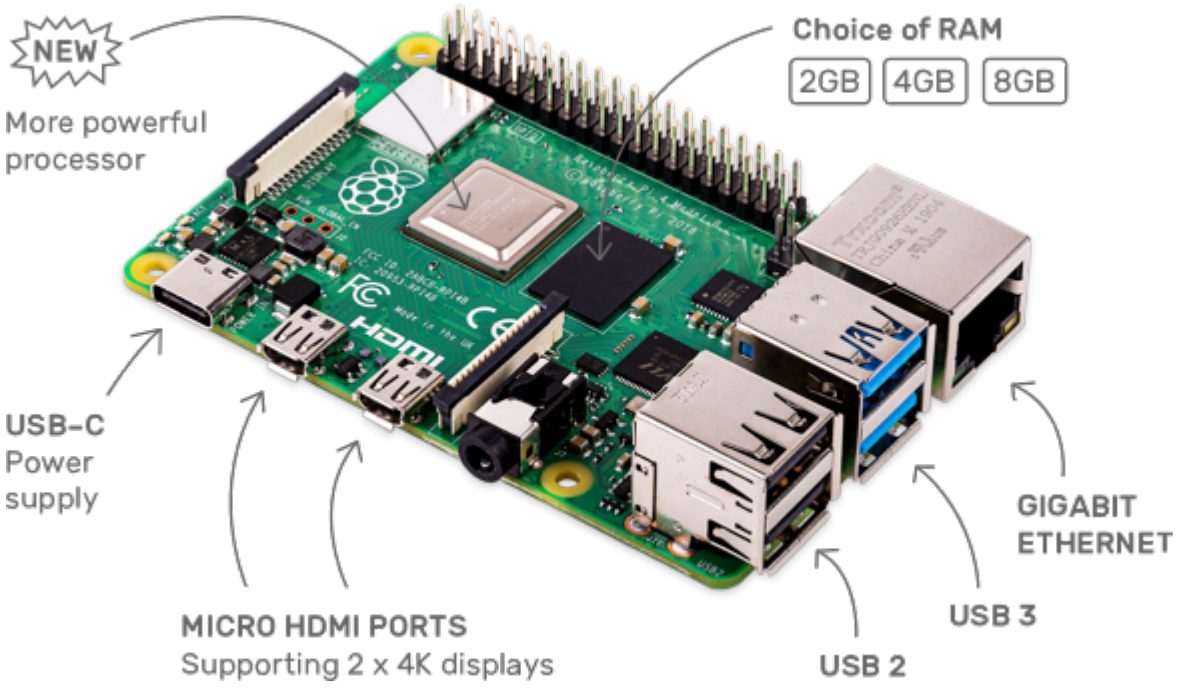
lus/).



Queste le caratteristiche:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

Raspberry Pi 4 Model B è l’ultima versione di Raspberry (https://www.raspberrypi.org/products/raspberry-pi-4-model-b/).



Queste le caratteristiche:

- Processore
  1.5GHz quad-core 64-bit ARM Cortex-A72 CPU (about 3x performance)
- 1GB,
  2GB o 4GB di LPDDR4 SDRAM
- Gigabit
  Ethernet
- Dual-band
  802.11ac wireless rete
- Bluetooth
  5.0
- 2
  porte USB 3.0 e 2 porte USB 2.0
- Supporto

dual monitor, con risoluzione fino a 4K

- VideoCore
  VI Graphics supporta OpenGL ES 3.x
- 4Kp60
  hardware decode di HEVC video
- Compatibilità
  con i precedenti prodotti Raspberry Pi

Ed infine la libreria to Pi4J (https://pi4j.com/1.2/index.html).

Questa
libreria ha lo scopo di fornire un'API di I/O orientata agli oggetti implementata
per i programmatori Java per accedere alle funzionalità di I/O complete della
piattaforma Raspberry Pi. Questo progetto astrae l'integrazione nativa di basso
livello e il monitoraggio degli interrupt per consentire ai programmatori Java
di concentrarsi sull'implementazione della logica di business dell'applicazione.

Come funziona il progetto:

Il progetto è suddiviso in 2 gruppi, il server e il client.

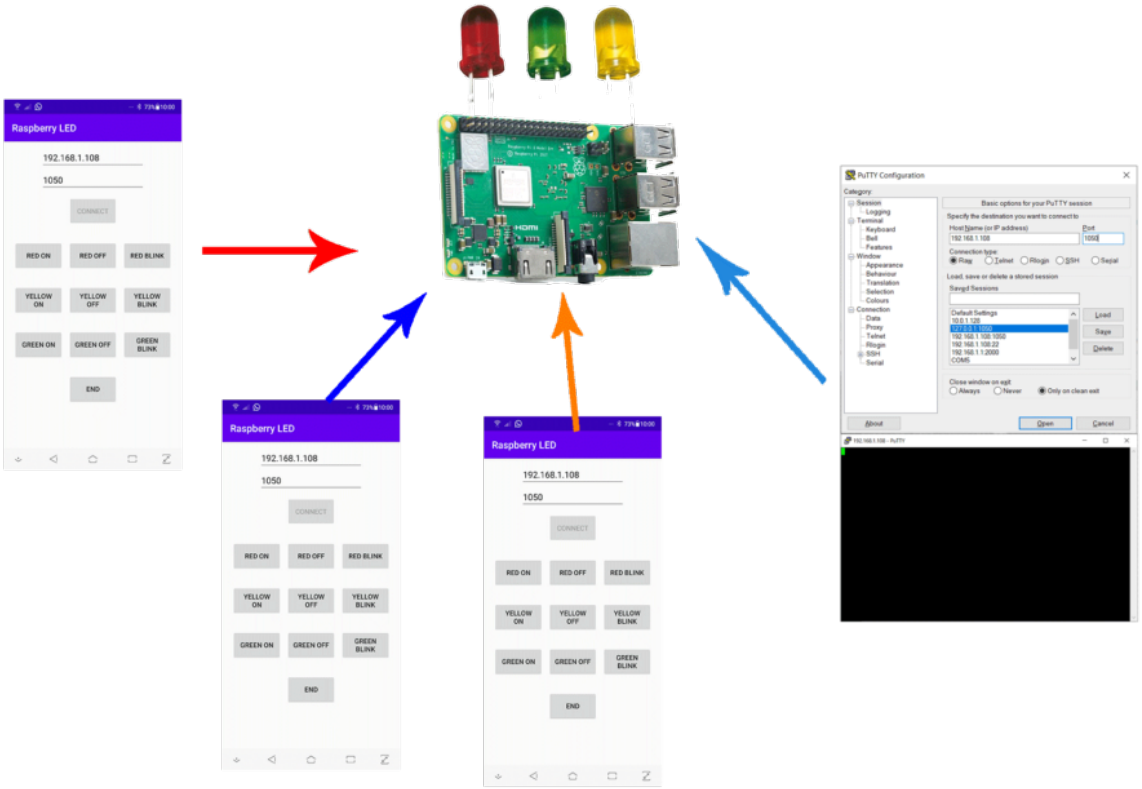
Il server viene realizzato tramite Raspberry sul quale viene fatto girare il Server TCP/IP Multithreading.

Il server in base a semplici comandi tipo “ON RED”, “OFF RED”, “BLINK RED”, “ON YELLOW”, “OFF YELLOW”, “BLINK YELLOW”, “ON GREEN”, “OFF GREEN”,
“BLINK GREEN” tutti ovviamente senza virgolette, accende, spegne e fa
lampeggiare i vari LED collegati alle porte GPIO di Raspberry.

Il client invece è realizzato tramite Android Studio.

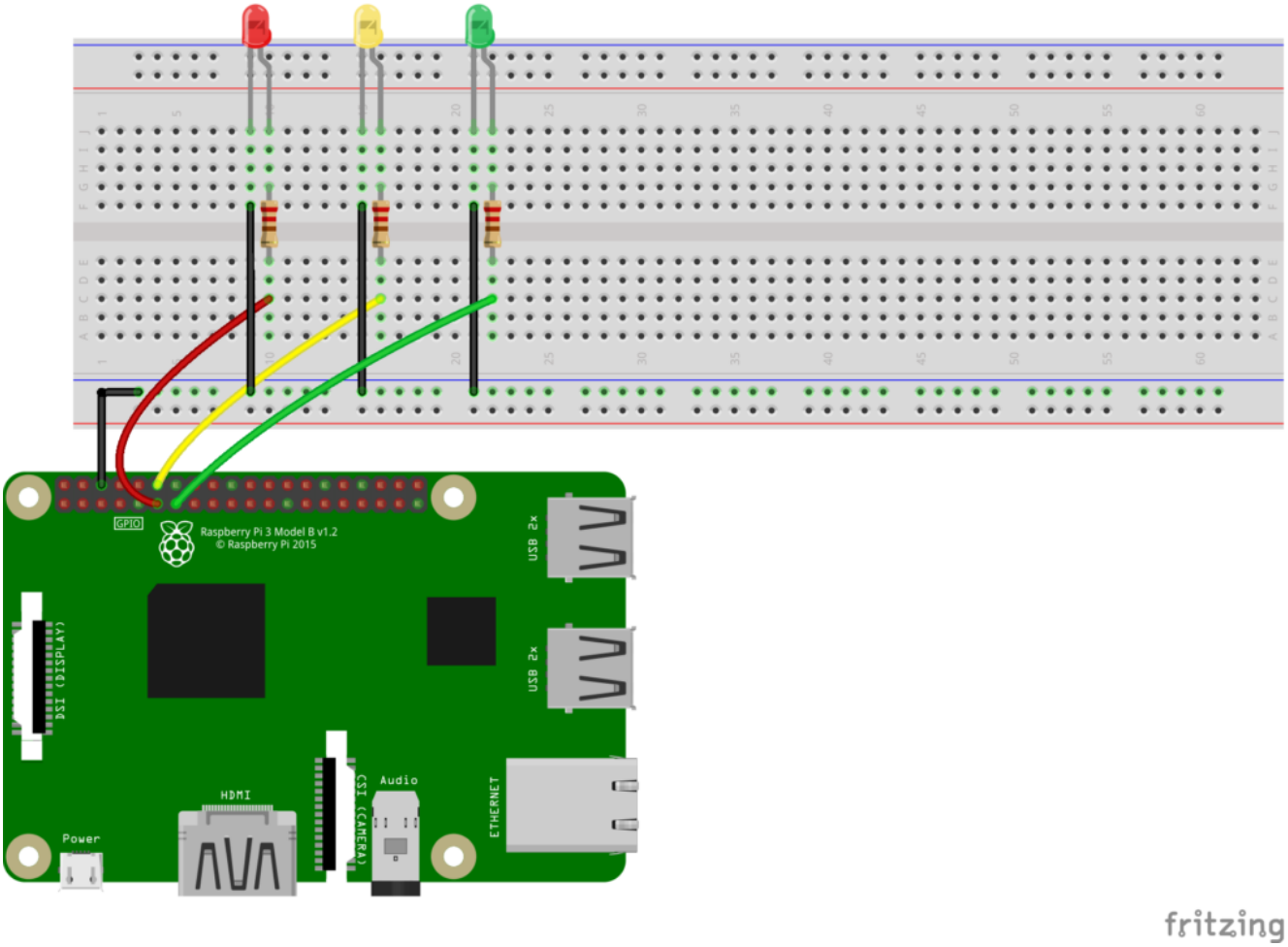
Il client ha una Activity contenente due Editbox per
digitare l’indirizzo IP e la porta di funzionamento del server e 11 bottoni in
grado di eseguire la connessione, accendere, spegnere, far lampeggiare i LED e
disconnettersi dal server.

È possibile anche utilizzare Putty da un qualsiasi PC, connettersi in modalità RAW all’indirizzo di Raspberry e alla porta 1050.

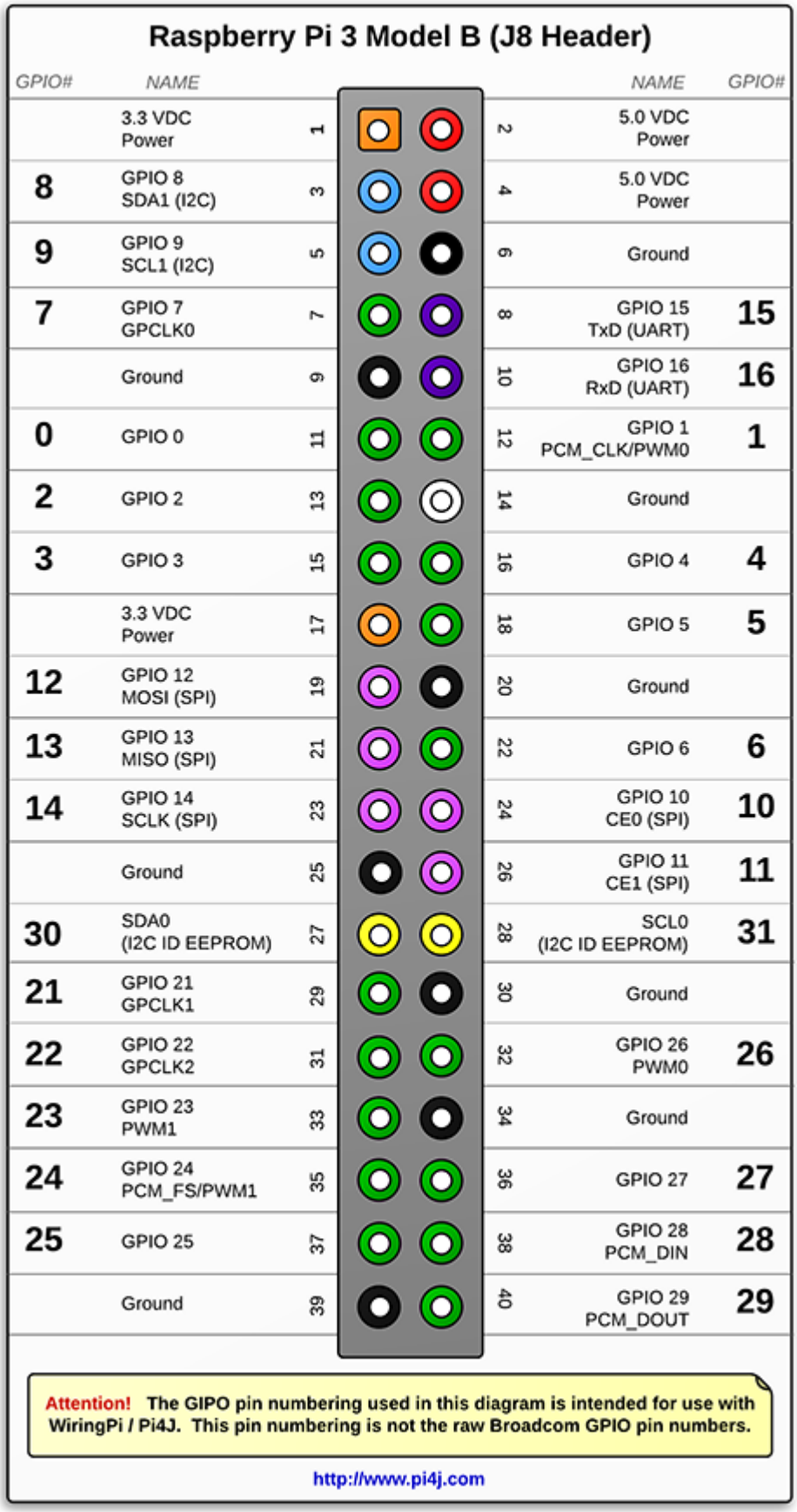


Schema di funzionamento

<u>Schema Server</u>:

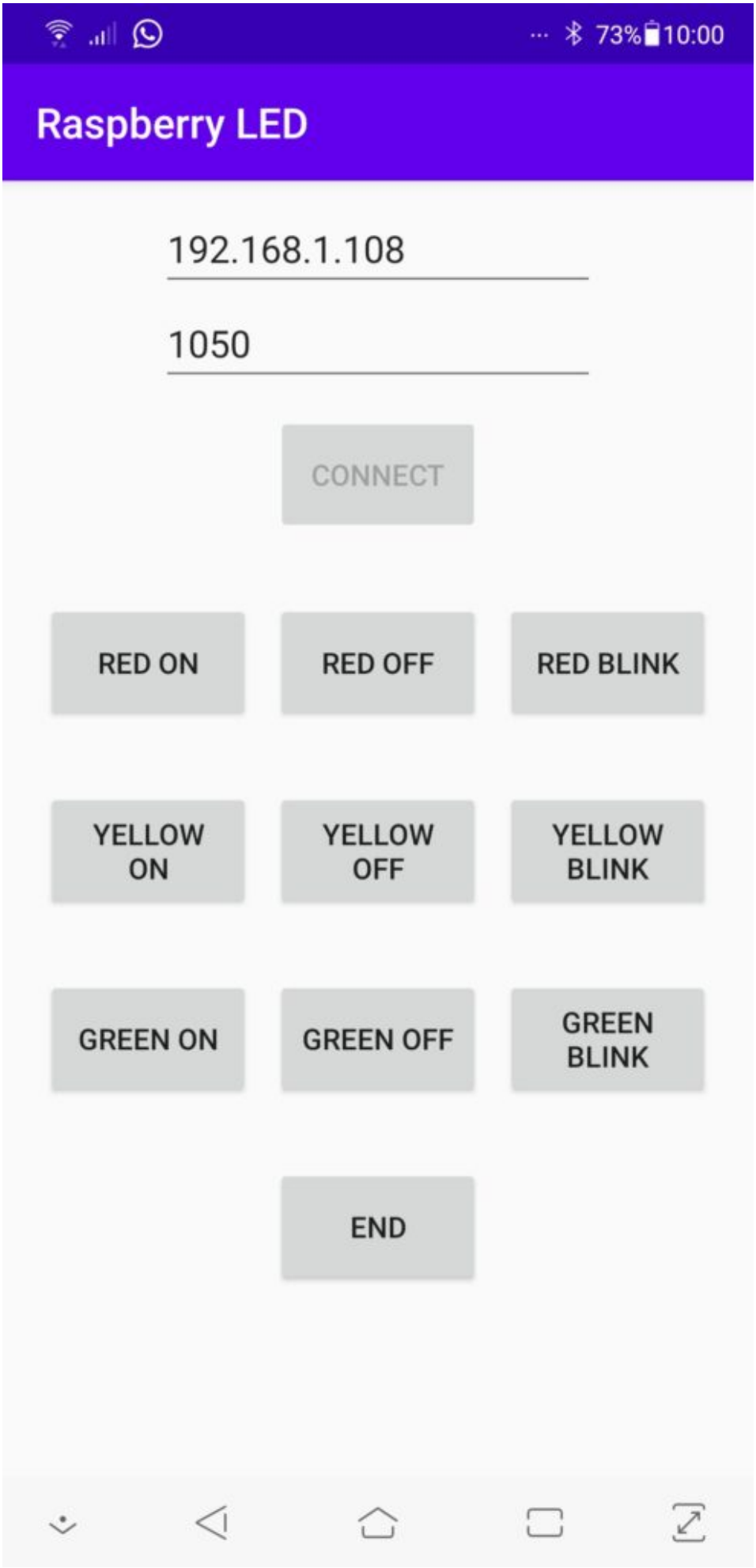


Schema di collegamento di Raspberry ai LED



Piedinatura delle GPIO di Raspberry

Schermata Client:


73% 10:00
Raspberry LED
192.168.1.108
1050
CONNECT
RED ON
RED OFF
RED BLINK
YELLOW ON
YELLOW OFF
YELLOW BLINK
GREEN ON
GREEN OFF
GREEN BLINK
END

Codice Sorgente:

Istruzioni installazione PI4J scrittura Server, compilazione ed esecuzione

Download Server TCP/IP Java

Download Client Android

Istruzioni per eseguire il server all’accensione di Raspberry come servizio