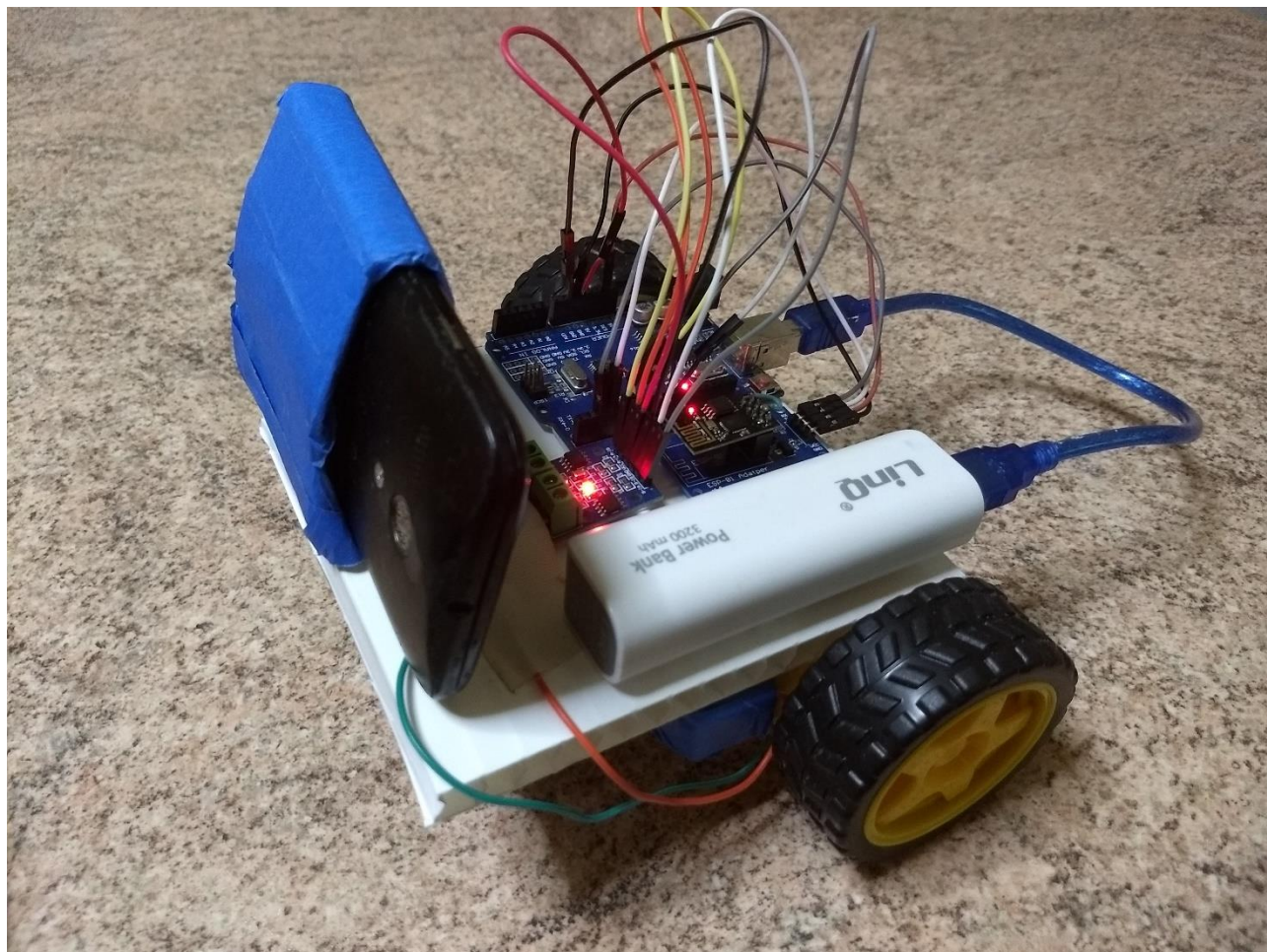


CREAZIONE DI UN VEICOLO CONTROLLABILE TRAMITE UN INTERFACCIA WEB



EMILIANO CARULLI

5° A INDIRIZZO INFORMATICA E TELECOMUNICAZIONE

ANNO SCOLASTICO 2017/2018

INDICE:

Introduzione:	pagina 3
Componenti/applicazioni utilizzate:	pagina 4
Schema elettrico:	pagina 5
Funzionalità:	pagina 6
Utilizzo sotto il punto di vista dell'utente:	pagina 7-8-9-10
Problemi e limitazioni ancora presenti:	pagina 10
Spiegazione di frammenti di codice sorgente che potrebbe essere poco chiari:	
	pagina 11-12
Conclusione:	pagina 13
Link componenti:	pagina 14

OBIETTIVO DA RAGGIUNGERE:

L'obiettivo del progetto è creare un veicolo controllabile tramite un'interfaccia, sfruttando le conoscenze acquisite durante il percorso di studi ricevuti dalla scuola superiore di secondo grado del "Mattei" di Vasto con specializzazione Informatica e Telecomunicazioni.

NASCE DA:

L'idea di creare un progetto fruendo di tutti i componenti già in possesso, comprati precedentemente per la progettazione di uno skateboard elettrico basilare; dopo grande riflessione si è deciso di crearne uno nuovo più articolato, in modo tale da mettere in pratica le conoscenze e le competenze acquisite nel percorso formativo, soffermandosi sugli argomenti che hanno suscitato maggiori stimoli.

La realizzazione e lo sviluppo del progetto sono stati possibili grazie al supporto di docenti e consultazioni di molteplici dispense digitali, piattaforme, blog e comunità online.

COMPONENTI IMPEGATI:

Strumenti usati:

Un computer (per compilare e caricare il codice Arduino)

Cavi vari (sono stati utilizzati due tipologie di cavi, la prima femmina-femmina e seconda maschio-maschio)

Saldatore (per saldare i pin dei motori DC e i pin del modulo ESP8266)

Coltello multiuso (per tagliare la struttura)

Righello (per misurare le dimensioni della base)

Tali strumenti sono stati utilizzati per la produzione della struttura del robot, l'assemblaggio, il collegamento di tutti i componenti elettronici e infine per la compilazione-caricamento del codice Arduino.

Componenti elettronici utilizzati:

Arduino Uno (x1)

Modulo ESP8266 (x1)

Motoriduttore DC con ruota(x2)

Modulo ponte H (x1)

Adattatore per modulo esp8266(x1)

Alcuni cavi ponticello (xN)

Banca di alimentazione USB 5V (x1)

Puoi trovare tutti i componenti online sul tuo negozio preferito (puoi trovare i link di tutti i componenti utilizzati nell'ultima pagina).

Applicazioni utilizzate:

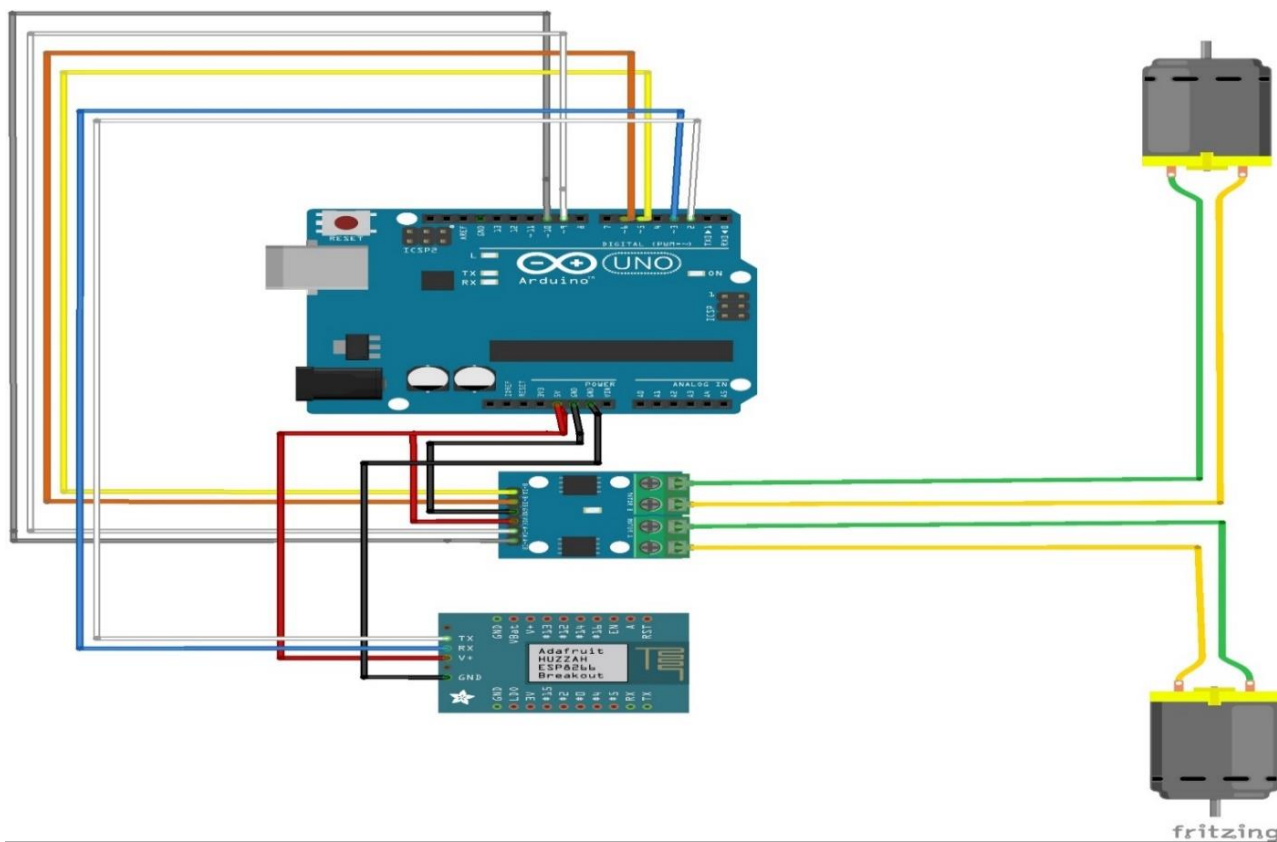
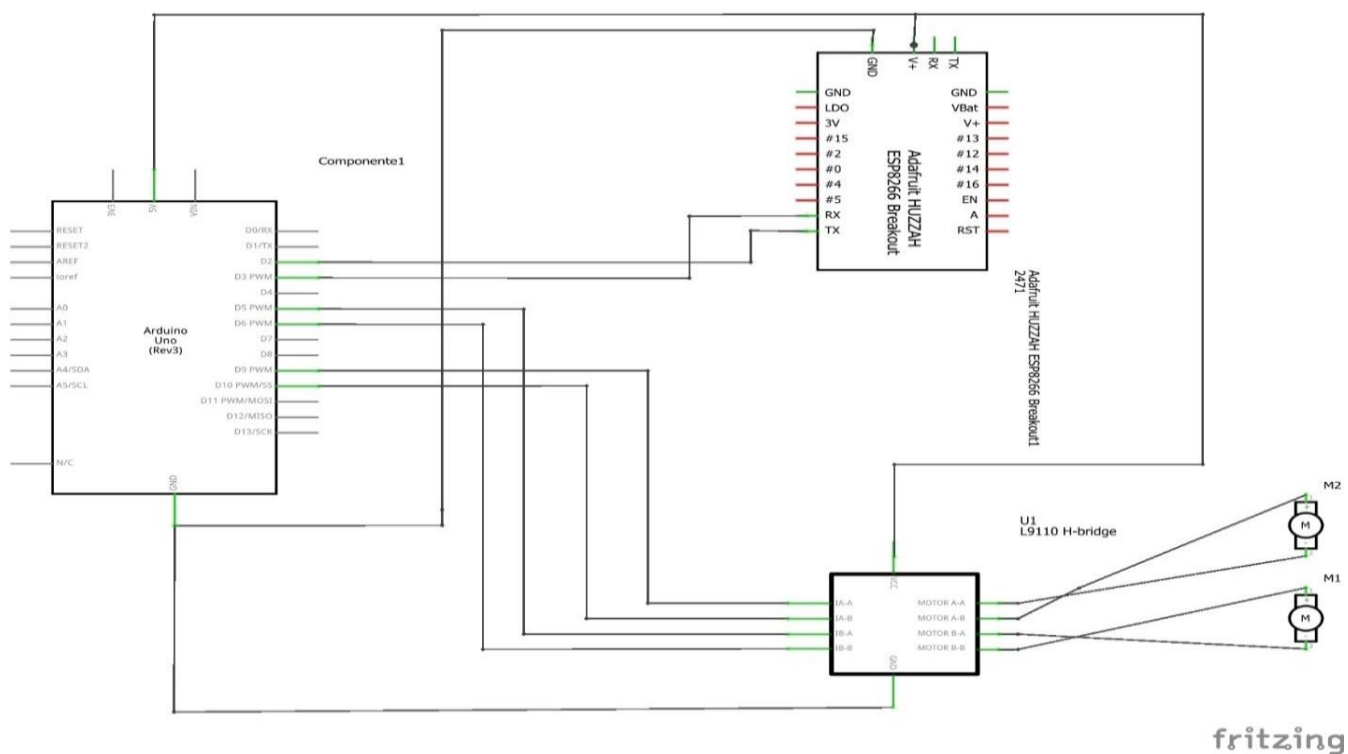
Arduino IDE: software open-source che semplifica la scrittura del codice e contemporaneamente permette il caricamento sulla scheda programmabile.

Fritzing.0.9.3B: software open-source che permette la progettazione di schemi elettrici.

IP Webcamera: programma gratuito scaricabile da Play Store, consente di trasformare il proprio telefono in una webcamera.

SCHEMA ELETTRICO:

Il circuito qui proposto utilizza un Arduino Uno come controller principale; esso ha sia il compito d'interfacciarsi con un modulo ESP8266, adatto per implementare la comunicazione Wi-Fi e sia il compito di controllare i motori DC, tramite aiuto di un ponte H-bridge. Per alimentare il circuito è stato utilizzato un normale power bank, in quanto rappresenta una soluzione ottimale per l'alimentazione dell'intero progetto, infatti esso può essere ricaricato in maniera del tutto indipendente e all'occorrenza sostituito con un altro di capacità maggiore.



*si noti che nello schema elettrico la rappresentazione dell'adattatore esp8266 utilizzato nel progetto, non è stata possibile in quanto i mezzi a disposizione non lo permettevano.

Funzionalità:

La funzionalità primaria del progetto è quella di poter controllare un robot a distanza in qualsiasi area geografica in cui sia possibile crearsi una connessione dati stabile, per realizzare tale progetto ci si è serviti di vari componenti come ad esempio i due motori DC; essendo degli attuatori essi necessitano di una corrente esterna per funzionare; questa tipologia di componenti utilizzati necessitano di una tensione continua variabile da un minimo di 3v a un massimo di 6v; nel progetto proposto per questione di comodità si è scelto di fornire una tensione pari a 5v (tensione massima che Arduino può erogare ai suoi componenti); successivamente, sul codice si è impostata una velocità predefinita che consente un controllo ottimale del robot; si è scelto di utilizzare questa tipologia di motori, in quanto rappresentano una soluzione ideale per le piattaforme robotizzate, infatti grazie alle dimensioni del motore (70 x 22 x 18) e quelle delle gomme (65 x 27 mm) si ha un buon compromesso fra potenza e grandezza. Successivamente essi saranno controllati tramite all'Arduino che si servirà di un ponte H-bridge in modo tale da far eseguire i comandi previsti nei 5 stati impostati.

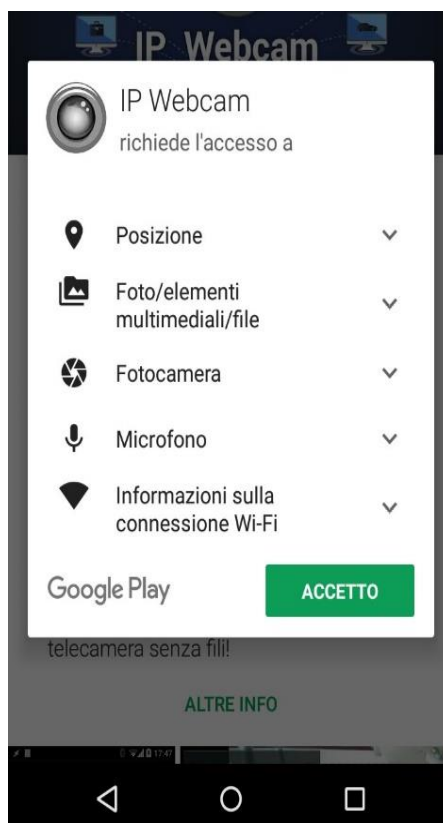
Il passaggio dei vari stati creati del progetto è stato implementato grazie al modulo esp8266, più precisamente modello AZDeliveryesp826601 con dimensioni 15,8 x 12,8 x 2,6 cm, con la peculiarità principale di avere un costo molto contenuto rapportato alle buone performance mostrate durante la comunicazione che avviene dalla pagina HTML (costruita grazie a uno spazio concesso dalla piattaforma Altervista) a tale modulo; si noti che esso è stato integrato nel progetto, tramite un adattatore che permette di risolvere una serie di problematiche presenti nella comunicazione seriale(esp-01). Infine si ponga attenzione sulla funzionalità che permette di vedere cosa si presenta davanti al robot, in tempo reale; si prenda in considerazione che tale funzionalità non era stata pensata durante la fase iniziale del progetto ma aggiunta in un secondo momento. Proprio per questo motivo ci si è trovati in un punto, in cui, vista la complessità e il tempo ridotto a disposizione ci si è dovuti accontentare di sfruttare una applicazione software reperibile direttamente sul Play Store; garantendosi in maniera semplice la trasformazione del proprio telefono in una webcam di rete, capace di trasferire il proprio contenuto su un apposito spazio web.

UTILIZZO SOTTO IL PUNTO DI VISTA UTENTE:

La funzionalità del progetto come abbiamo premesso nella prima parte, dipende strettamente dalla presenza di una linea dati stabile, che nella maggior parte dei casi sarà emessa dallo smartphone utilizzato come camera istantanea. È necessario che sia il modulo esp8266, sia il computer utilizzato per controllare il progetto siano connessi alla stessa rete, nel caso questo non dovesse accadere, la funzionalità dell'intero progetto sarà completamente compromessa; si noti che non è obbligatorio avere sempre presente una connessione dati alla mano; infatti, con opportune modifiche al codice sorgente dell'Arduino possiamo sfruttare anche una qualsiasi rete LAN presente nei dintorni (a patto che ne abbiamo il consenso), una volta connessi entrambi i dispositivi alla rete, non resta altro da fare che accedere alla pagina web con indirizzo URL" <http://carulli.altervista.org/Contollo.php>"; fatto ciò ci ritroveremmo una pagina HTML che necessita dell'indirizzo IP della webcam e l'indirizzo IP del modulo esp8266, possiamo vedere che queste informazioni, sono inserite di default; si noti, però, che al variare della rete di appartenenza in maniera del tutto automatica, cambieranno la copia d'indirizzi IP da utilizzare. Per ottenere i propri indirizzi sarà necessario per prima cosa scaricare dal Play Store l'applicazione IP Webcam:



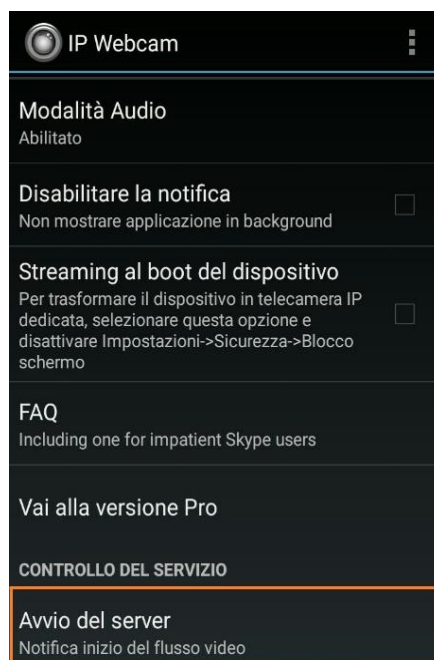
Successivamente accettare le autorizzazioni chieste, in quanto sono tutte necessarie per sfruttare a pieno le funzionalità dell'applicazione:



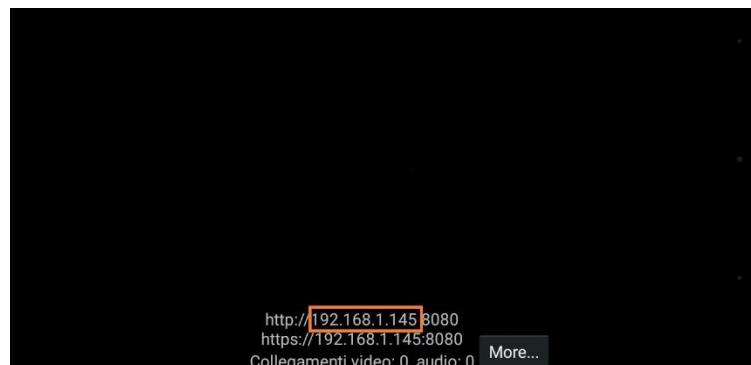
successivamente aprire l'applicazione:



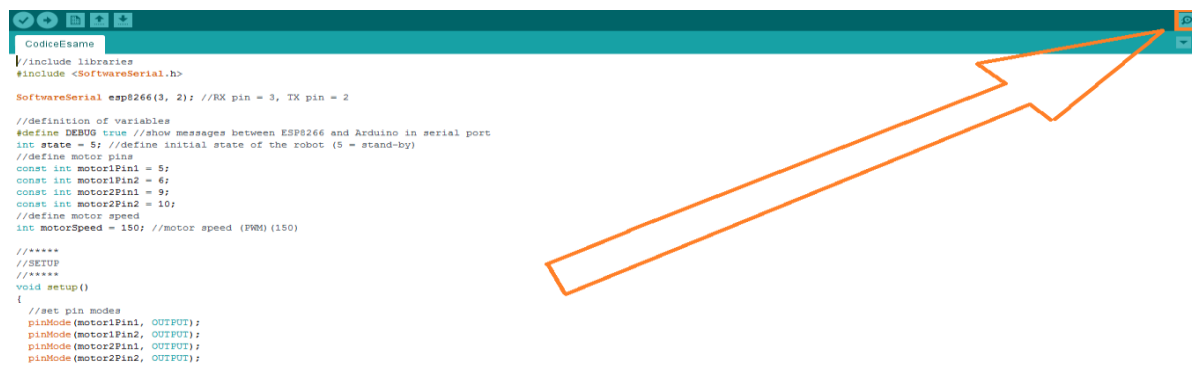
ora andare sull'ultima voce del menu (Avvio del server):



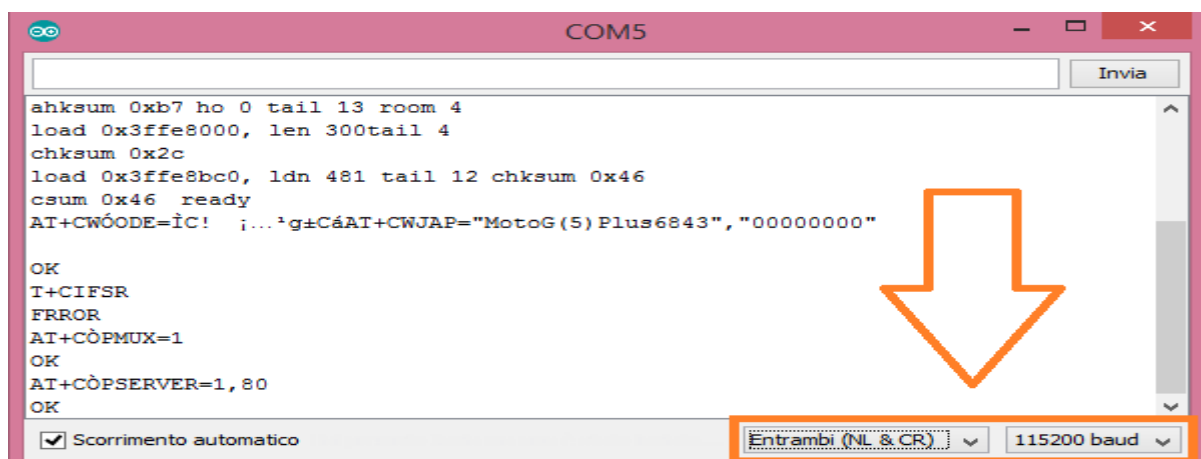
e leggere l'indirizzo IP ottenuto, nel caso si avessero dei dubbi ricordiamo che esso è formato da quattro gruppi divisi da un punto, in questo caso "192.168.1.145".



Per quanto riguarda, invece, ottenere l'indirizzo IP del modulo esp8266 la procedura è più complessa in quanto bisogna accedere al codice sorgente dell'Arduino, avendo contemporaneamente il progetto connesso tramite cavo USB e cliccare sulla spunta presente in alto a destra (a forma di lente d'ingrandimento):



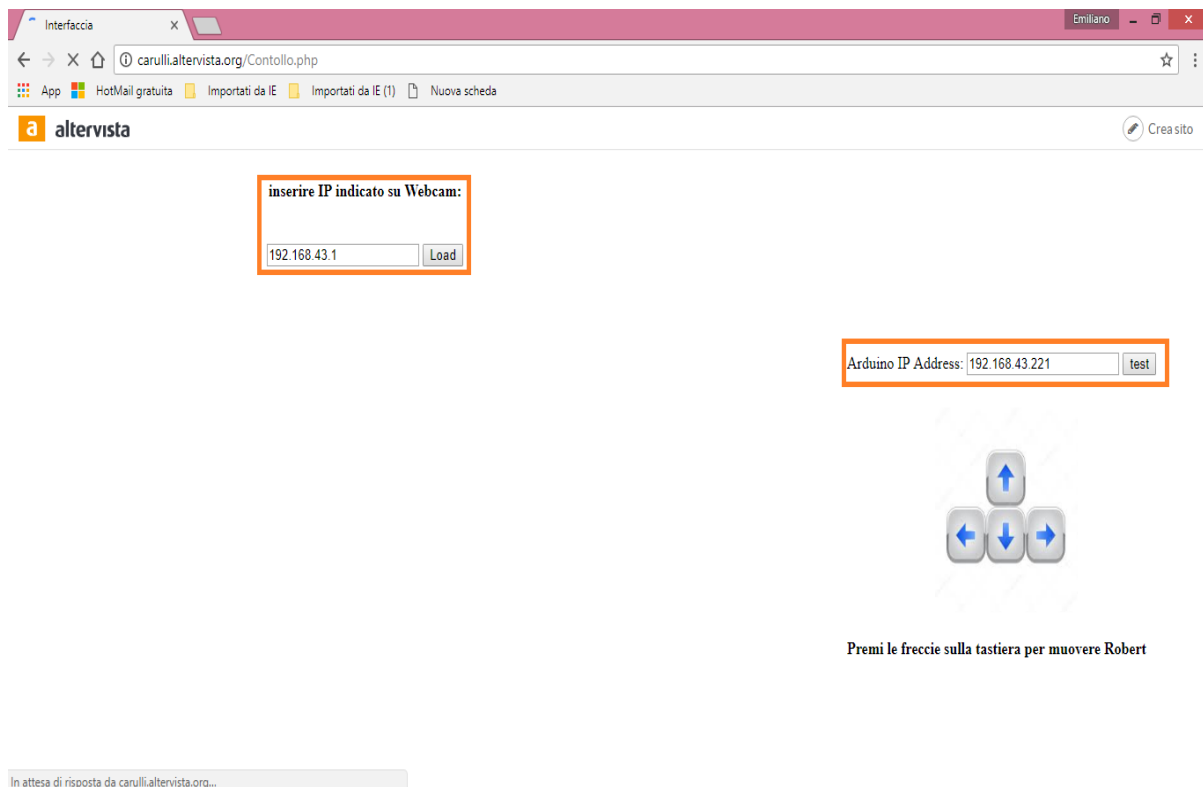
una volta apparsa la nuova finestra, impostare la velocità dei baud a 115200 e la comunicazione bidirezionale:



aspettare fin quando non compaiono una serie d'informazioni dove potremmo trovare il nostro IP:



ora basta staccare il cavo USB e collegarlo alla fonte esterna scelta, in questo caso un power bank; tornare sulla pagina web precedentemente aperta, inserire i due indirizzi IP richiesti nelle caselle di testo, premere i pulsanti presenti sulla pagina e cominciare a controllare il robot tramite i tasti direzionali.



PROBLEMI E LIMITAZIONI ANCORA PRESENTI:

Si noti che il progetto presentato, se pur funzionante in ogni suo aspetto ha delle problematiche dovute ai vari componenti di natura economica scelti, come il fatto di ripetere più volte lo stesso comando al fine di far cambiare direzione al progetto. Il fenomeno è dovuto al fatto che i segnali trasmessi non siano gestiti nella maniera prevista, per quanto riguardano le limitazioni, invece sono due: la prima è che il controllo del progetto può essere fatto solamente nel raggio d'azione della rete fornita dal telefono che funge da Hotspot e il secondo è che il controllo del progetto può essere fatto solamente da un terminale che possiede una tastiera fisica.

SPIEGAZIONE CODICE:

Di seguito sono riportati diversi screenshot con opportuna spiegazione che permettono di capire meglio cosa si è fatto nel codice sorgente (sarà allegata a questa documentazione l'intero codice opportunamente commentato).

Premesso che non sia stata necessaria alcuna libreria per la comunicazione con il modulo ESP-8266 in quanto si sono sfruttate solo i seguenti comandi AT ordinari (una serie di istruzioni definite sul firmware predefinito ESP8266), ci si è dovuti servire della funzione ausiliaria (sendData) che permette l'invio di dati, tramite la porta seriale (da Arduino a ESP8266):

```
sendData("AT+RST\r\n", 2000, DEBUG);
sendData("AT+CWMODE=1\r\n", 1000, DEBUG);
sendData("AT+CWJAP=\"XXXXX\", \"YYYYY\"\r\n", 2000, DEBUG);
sendData("AT+CIFSR\r\n", 1000, DEBUG);
sendData("AT+CIPMUX=1\r\n", 1000, DEBUG);
sendData("AT+CIPSERVER=1,80\r\n", 1000, DEBUG); |
```

Fatta questa premessa dunque è necessario, per prima cosa, la creazione di una porta seriale che consenta la comunicazione tra Arduino e ESP8266; questo è stato fatto tramite la libreria SoftwareSerial, utilizzando i pin digitali 2 e 3 come possiamo vedere di seguito:

```
#include <SoftwareSerial.h>

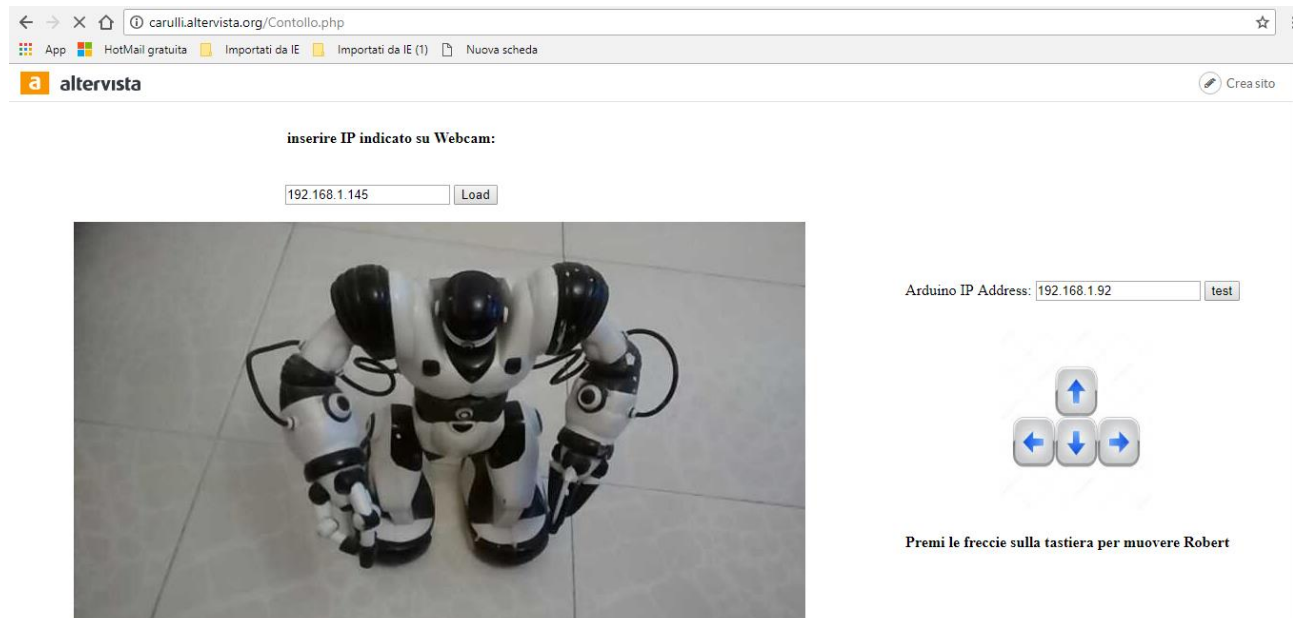
SoftwareSerial esp8266(3, 2); //RX pin = 3, TX pin = 2
```

L'uso del codice sotto indicato, invece, dà la capacità ad Arduino di ricevere, sulla porta seriale, cosa stia avvenendo nel momento in cui i segnali siano ricevuti correttamente, così da poter sempre vedere cosa stia realmente ricevendo Arduino e verificare le relative risposte emesse:

```
if (esp8266.available())
{
  if (esp8266.find("+IPD, "))
  {
    String msg;
    esp8266.find("?");
    msg = esp8266.readStringUntil(' ');
    String command = msg.substring(0, 3);
    Serial.println(command);
  }
}
```

Da questo momento l'Arduino può passare in cinque diversi stati (da cm1 a cm5). Ogni volta che esso riceverà uno di questi comandi, entrerà in uno dei cinque possibili stati (andando avanti, spostandosi indietro, spostandosi a destra, spostandosi a sinistra e rimanendo in attesa), mantenendo lo stato assegnato, finché non riceverà un comando diverso.

Per quanto riguarda l'interfaccia HTML progettata per controllare il robot, vengono utilizzati due moduli di testo che permettono d'immettere l'indirizzo IP del modulo esp8266 e l'indirizzo IP del server video / audio (dall'app Android IP Webcam). Come si può leggere dalla schermata, il controllo sarà possibile dai tasti direzionali della tastiera, infatti si è pensato che questa scelta favorisca "l'intuito" dell'utente che automaticamente assocerà le direzioni presenti sulla tastiera, alle direzioni che potrà assumere il robot.



Scendendo più nel dettaglio, si può notare che tale pagina è stata divisa in due porzioni, entrambe aventi una casella di testo assegnata, quest'ultime saranno destinate ad ospitare li l'indirizzi IP necessari per stabilire sia la comunicazione video/audio, sia la comunicazione con il modulo esp8266; come si potrà vedere dal codice sorgente questa semplice pagina HTML richiamerà a sua volta due moduli JavaScript:

```
<html xmlns="http://www.w3.org/1999/xhtml">
</head>
  <title>Interfaccia</title>
  <!-- richiamo i due file JavaScript -->
  <script src="jquery.js"></script>
  <script src="myscript.js"></script>
</head>
```

il primo è chiamato "myscript.js" e il secondo ha il nome "jquery.js". Il primo modulo è stato scritto interamente dall'alunno e consente di gestire le chiamate asincrone, come in questo caso il caricamento dei contenuti dinamici (webcamera e comunicazione con il modulo esp8266). Il secondo modulo, invece, non è stato scritto dal sottoscritto in quanto esso è una vera e propria libreria per la gestione delle applicazioni web. Essa infatti nasce con l'obiettivo di semplificare la selezione, la manipolazione e la gestione degli eventi in pagine HTML (nonché implementare funzionalità AJAX); il modulo jquery è reperibile gratuitamente sotto i termini della Licenza MIT. Attualmente diversi articoli hanno fatto notare come, tale modulo stia diventando una libreria indispensabile per svolgere

determinati compiti con JavaScript; non a caso nelle versioni più recenti di JavaScript e dei browser, le funzionalità di jquery sono diventate funzionalità standard di default.

Conclusione:

Dopo aver terminato il progetto e avendolo testato nei vari ambiti, possiamo concludere che per gli impieghi a esso riservati, garantisce un buon livello di funzionalità, anche se sono ancora presenti dei piccoli difetti di funzione che saranno risolti in futuro; una volta che il progetto avrà raggiunto un livello di qualità superiore a quello attuale si avrà l'intenzione di renderlo pubblico, attraverso comunità online e blog così da fornire già un problema risolto a tutti coloro che vogliano affrontare questo tipo di lavoro, lasciando piena libertà a tutti coloro che ne vogliano implementare le capacità, tramite integrazione di nuovi moduli e sensori, ciò in quanto si è capito da diverso tempo che tale comportamento è alla base della crescita esponenziale nel mondo digitale.

LINK COMPONENTI:

Motori DC:

https://www.amazon.it/KKmoon-Motoriduttori-Ruota-Arduino-Project/dp/B07B4875JJ/ref=sr_1_7?ie=UTF8&qid=1527929950&sr=8-7&keywords=motori+dc+arduino

Modulo esp8266:

https://www.amazon.it/AZDelivery-esp8266-Arduino-Raspberry-Microcontroller/dp/B01LK83TX0/ref=sr_1_3?ie=UTF8&qid=1527929995&sr=8-3&keywords=modulo+esp8266

Adattatore esp8266:

https://www.amazon.it/ESP8266-seriale-wireless-adattatore-compatibile/dp/B01LZLVEJ4/ref=pd_sim_147_4?encoding=UTF8&pd_rd_i=B01LZLVEJ4&pd_rd_r=5541af12-6643-11e8-8704-c10a6af95ec8&pd_rd_w=qYqcf&pd_rd_wg=gWCH4&pf_rd_i=desktop-dp-sims&pf_rd_m=A111L2PNWYJU7H&pf_rd_p=6946954921869226644&pf_rd_r=PPD1T5706SCKR3YT5QW2&pf_rd_s=desktop-dp-sims&pf_rd_t=40701&psc=1&refRID=PPD1T5706SCKR3YT5QW2

Cavi femmina femmina:

https://www.amazon.it/femmina-femmina-cavetti-raspberry-arduino-Arduino/dp/B01GI3X1XK/ref=sr_1_4_sspa?s=electronics&ie=UTF8&qid=1527930317&sr=1-4-spons&keywords=cavi+femmina+femmina&psc=1

Cavi maschio maschio:

https://www.amazon.it/gp/product/B00KBPQYO0/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1

Arduino:

https://www.gearbest.com/arduino-scm-supplies/pp_311407.html?wid=1433363

H-bridge:

<https://grabcad.com/library/motor-driver-l9110-1>