



TITOLO	NAO e Arduino si parlano!	
PRODOTTI UTILIZZATI	 ALDEBARAN <small>= SoftBank Group</small>	 ARDUINO
OBIETTIVO	<p>Far comunicare una scheda Arduino con il robot umanoide NAO. L'utente chiederà vocalmente a NAO di accendere/spengere un LED collegato alla scheda Arduino, oppure di sapere qual è il valore di temperatura o luminosità di due sensori collegati ad Arduino. NAO dovrà interpretare i comandi vocali dell'utente e fornire risposte.</p>	
AUTORE	Pietro Alberti (Media Direct srl, Italia) – p.alberti@mediadirect.it	
RINGRAZIAMENTI	Jonas Lerebours (Aldebaran Robotics, France)	
	<i>Dedicato alla mia famiglia: Marco, Luca, Chiara, Francesco e Maria</i>	



ABSTRACT

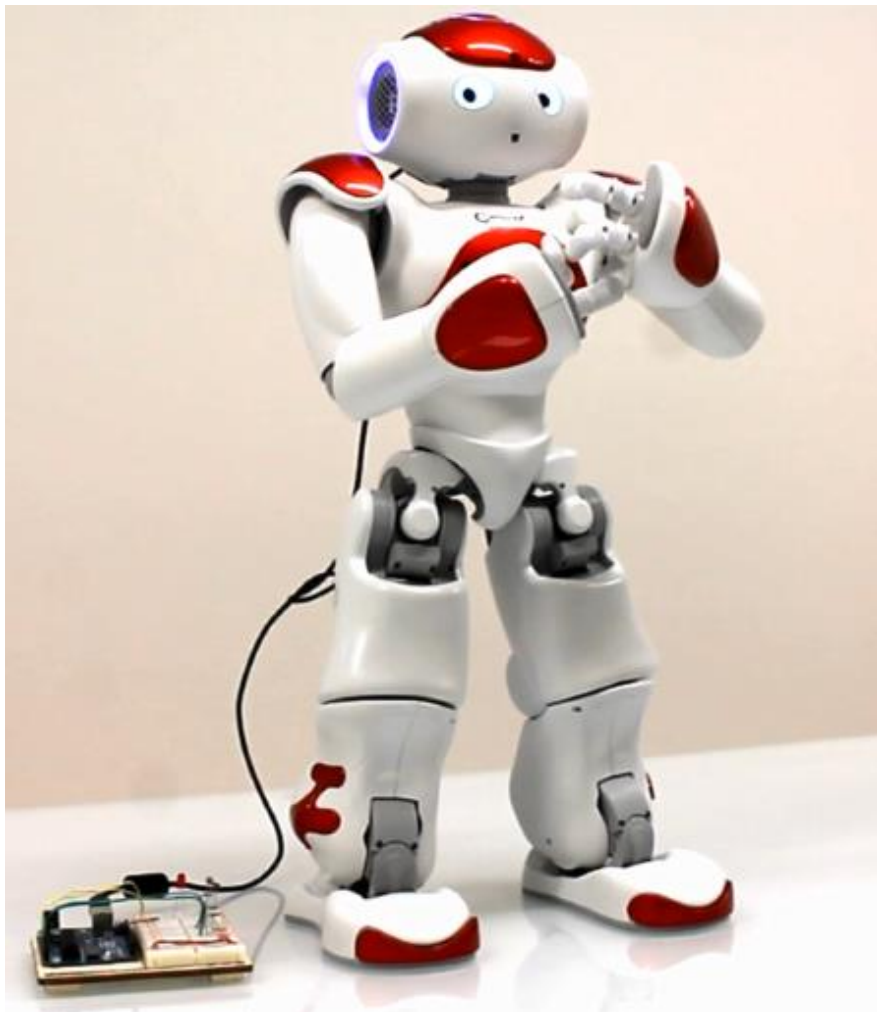
Che sfida interessante far comunicare tra loro due tra i prodotti per robotica più famosi al mondo: NAO e Arduino!

NAO ha una porta USB nella parte posteriore della testa a cui si può collegare direttamente una scheda Arduino.

L'utente chiederà a NAO di accendere/spegnere un LED e/o leggere i valori di due sensori di luminosità e temperatura collegati alla Arduino.

NAO, dopo aver capito i comandi dell'utente, invierà dei messaggi sulla seriale. Tali messaggi verranno interpretati da Arduino, che esaudirà le richieste di NAO, e risponderà sempre sulla seriale con altrettanti messaggi verso NAO.

I messaggi di ritorno a NAO implicheranno che il nostro piccolo grande amico dirà il valore di temperatura, di luminosità e/o se ha acceso/spento il LED.



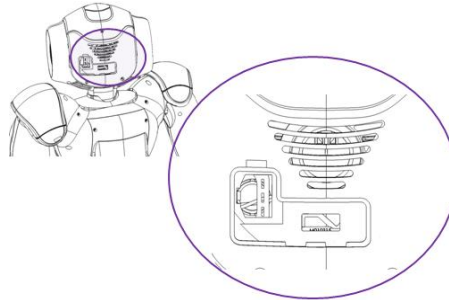
DESCRIZIONE DEL SISTEMA: NAO + ARDUINO

NAO: robot umanoide alto 58 cm in grado di camminare, parlare, ascoltare, collegarsi a Internet, corredato di molti sensori. Per una descrizione completa si rimanda al sito: <http://www.campustore.it/nao>

NAO possiede una **porta USB** nella parte posteriore della testa, normalmente usata per il reflashing suo sistema operativo. Celata da una copertura rimovibile, questa è una porta USB standard che permette di collegare dispositivi aggiuntivi, ad esempio il sensore Kinect o una mitica scheda Arduino!

Il sistema operativo NAOqi include i driver per i chipset FTDI, pertanto un dispositivo dotato di tale tipo di interfaccia quando collegato a tale porta USB di NAO viene visto come una **porta seriale** o TTY.

Quindi, se prendiamo una scheda Arduino Uno e la colleghiamo a NAO mediante un cavo USB, qualsiasi cosa inviata da Arduino sulla seriale verrà di fatto inviata a NAO, e così il gioco è fatto!



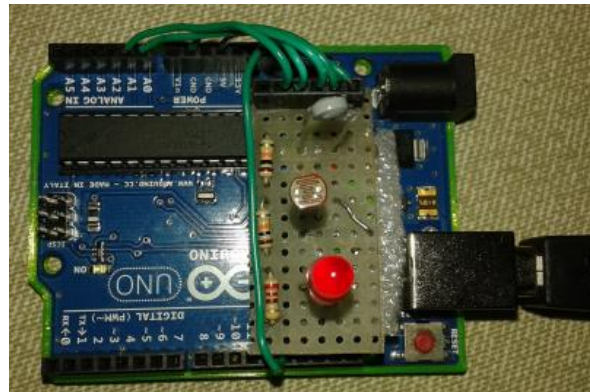
ARDUINO: la scheda di prototipazione più economica, più famosa, più...

Per una descrizione un po' più approfondita si rimanda al sito: www.campustore.it/arduino

In questa esperienza, ho utilizzato:

- Scheda Arduino Uno rev3
- 1 led rosso collegato al "solito" pin 13
- 1 termistore NTC collegato al pin A0
- 1 fotoresistore collegato al pin A1

All'inizio sono partito con una basetta dello starter kit, poi con fili volanti ed infine, per rendere più ordinato il tutto, ho saldato i 3 componenti e 3 resistenze su un pezzetto ritagliato di basetta millefori, appoggiando poi il tutto su un pezzetto di polistirolo, per isolare la scheda.



CAVO USB: classico "cavo stampante", che funge da cordone ombelicale tra NAO e Arduino, consentendo la comunicazione seriale.



DESCRIZIONE DEL FUNZIONAMENTO DEI PROGRAMMI SVILUPPATI

Il sistema si basa sullo scambio di messaggi tra NAO e Arduino usando il canale seriale.
 Ho sviluppato **2 programmi**: uno per NAO e uno per Arduino; entrambi usano la porta seriale.
 I due programmi sono commentati abbastanza.

Esempio di funzionamento:

- 1) L'utente dice a NAO: "temperatura"
- 2) NAO capisce il comando vocale e trasmette sulla porta seriale il messaggio "getTemp"
- 3) Arduino si accorge che nella seriale è stato scritto un messaggio e lo interpreta, andando a leggere il pin A0 e scrivendo sulla seriale il valore di temperatura
- 4) NAO legge il valore dalla seriale e lo comunica vocalmente all'utente

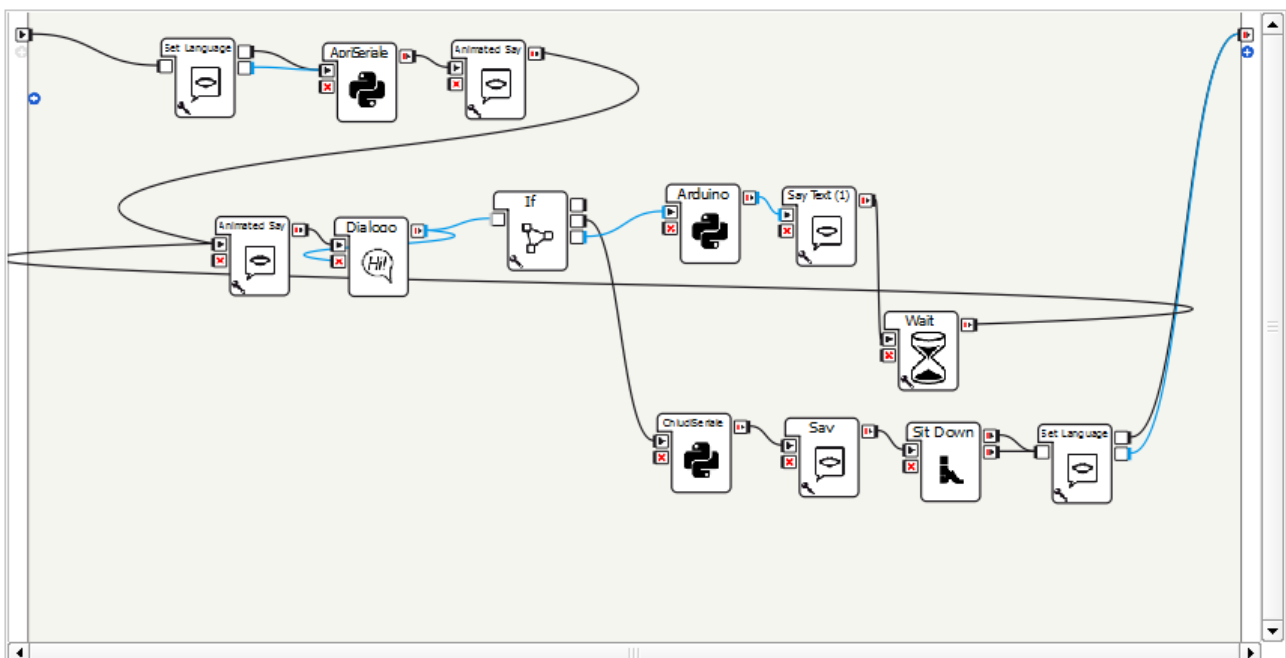
In modo analogo, l'utente dirà "luminosità", "accendi" o "spegni".

Tabella di messaggi:

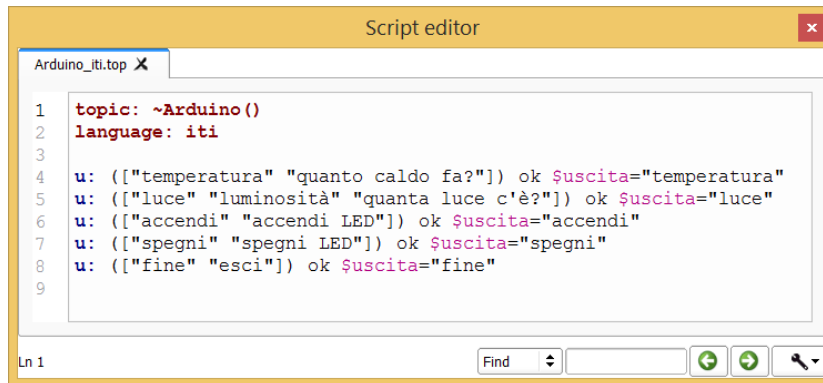
Parole che NAO può capire:	NAO passa ad Arduino:
"temperatura", "quanto caldo fa?"	getTemp
"luce", "luminosità", "quanta luce c'è?"	getLight
"accendi", "accendi LED"	setLed
"spegni", "spegni LED"	resetLed
"fine", "esci"	
	setLanguageITA
	setLanguageENG

Programma per NAO:

- Sviluppato in Choregraphe
- Sviluppato un blocco in Python per la comunicazione seriale, usando la relativa classe standard di NAOqi (import serial)
- Uso di un blocco di "Dialogo" per l'interazione utente-robot NAO



Dal blocco di dialogo escono delle stringhe che poi vengono passate al blocco If che segue: il terminale uscita è di tipo stringa e può assumere i valori seguenti (vedasi il dialogo 'Arduino_iti.top'): "temperatura", "luce", "accendi", "spegni", "fine".



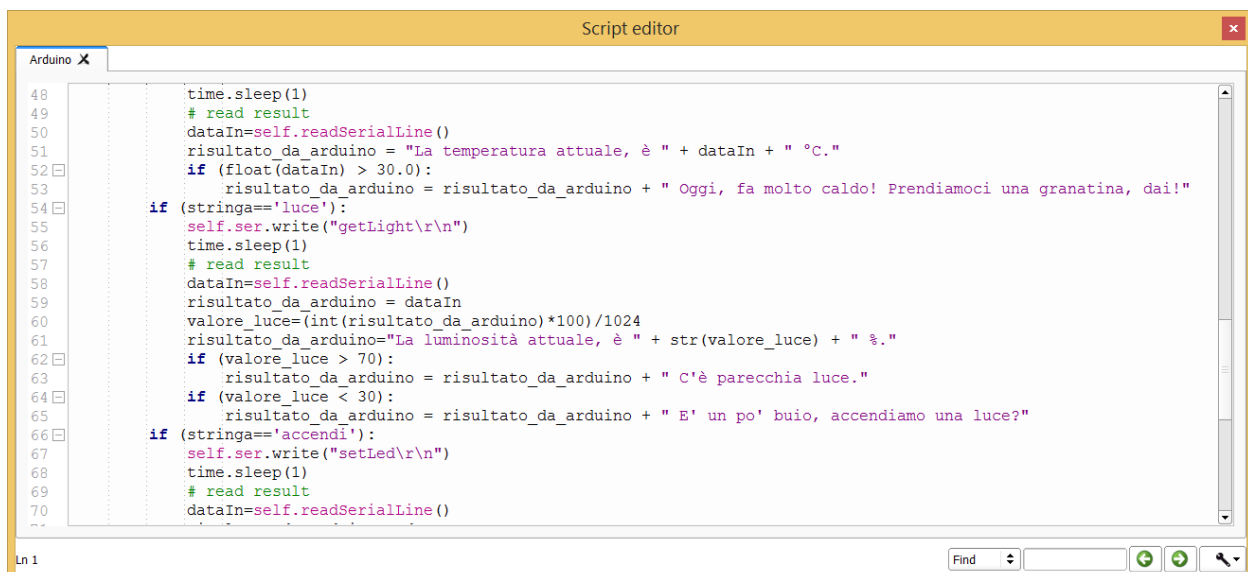
```

1  topic: ~Arduino()
2  language: iti
3
4  u: (["temperatura" "quanto caldo fa?"]) ok $uscita="temperatura"
5  u: (["luce" "luminosità" "quanta luce c'è?"]) ok $uscita="luce"
6  u: (["accendi" "accendi LED"]) ok $uscita="accendi"
7  u: (["spegni" "spegni LED"]) ok $uscita="spegni"
8  u: (["fine" "esci"]) ok $uscita="fine"
9

```

Nel blocco in Python di Choregraphe, dove NAO riceve dati da Arduino, ho messo qualche istruzione condizionale per far dire a NAO qualcosina in più, in base ai valori di temperatura e luminosità:

- Se la temperatura supera i 30 gradi, allora NAO dice "Oggi, fa molto caldo! Prendiamoci una granatina, dai!"
- Se è la luminosità supera il 70%, allora NAO dice "C'è parecchia luce."
- Se è la luminosità è minore del 30%, allora NAO dice "E' un po' buio, accendiamo una luce?"

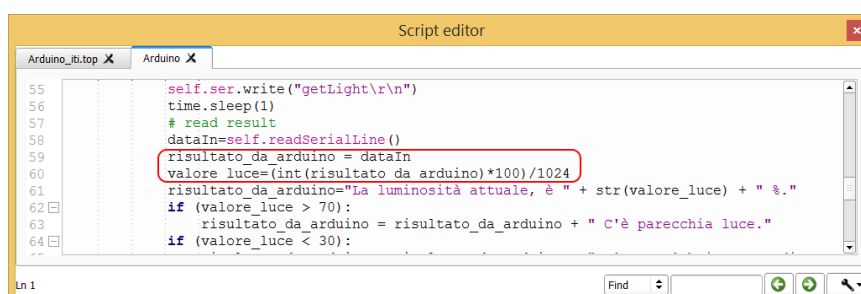


```

48  time.sleep(1)
49  # read result
50  dataIn=self.readSerialLine()
51  risultato_da_arduino = "La temperatura attuale, è " + dataIn + " °C."
52  if (float(dataIn) > 30.0):
53      risultato_da_arduino = risultato_da_arduino + " Oggi, fa molto caldo! Prendiamoci una granatina, dai!"
54  if (stringa=='luce'):
55      self.ser.write("getLight\r\n")
56      time.sleep(1)
57      # read result
58      dataIn=self.readSerialLine()
59      risultato_da_arduino = dataIn
60      valore_luce=(int(risultato_da_arduino)*100)/1024
61      risultato_da_arduino="La luminosità attuale, è " + str(valore_luce) + " %."
62  if (valore_luce > 70):
63      risultato_da_arduino = risultato_da_arduino + " C'è parecchia luce."
64  if (valore_luce < 30):
65      risultato_da_arduino = risultato_da_arduino + " E' un po' buio, accendiamo una luce?"
66  if (stringa=='accendi'):
67      self.ser.write("setLed\r\n")
68      time.sleep(1)
69      # read result
70      dataIn=self.readSerialLine()
71

```

Ultima cosa: il valore di luminosità proveniente dal fotoresistore collegato ad Arduino è un numero tra 0 e 1024, ove 0 significa luminosità minima/buio e 1024 significa luminosità massima. Tale valore lo mappo nell'intervallo 0-100, per avere un valore in percentuale (%).



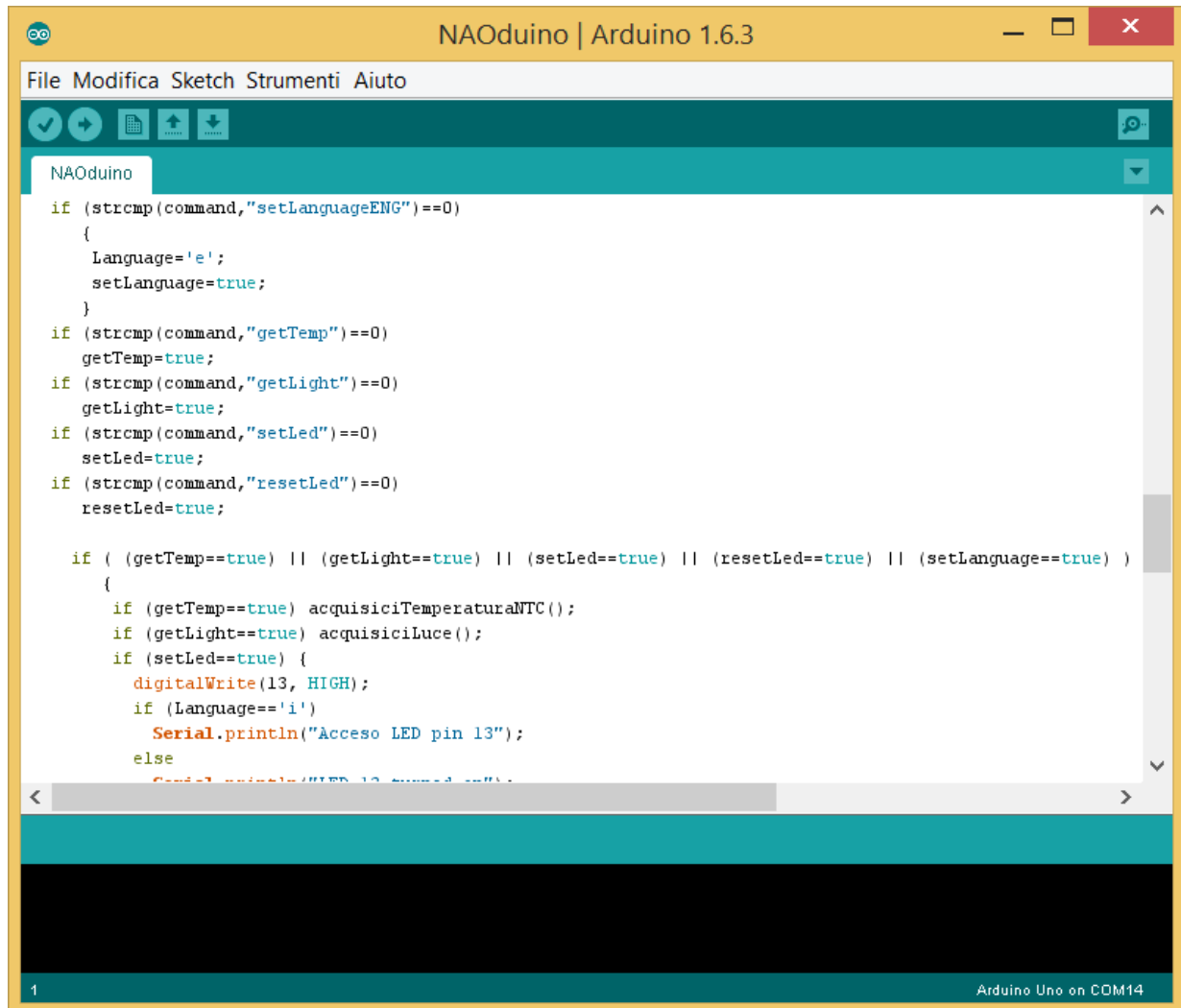
```

55  self.ser.write("getLight\r\n")
56  time.sleep(1)
57  # read result
58  dataIn=self.readSerialLine()
59  risultato_da_arduino = dataIn
60  valore_luce=(int(risultato_da_arduino)*100)/1024
61  risultato_da_arduino="La luminosità attuale, è " + str(valore_luce) + " %."
62  if (valore_luce > 70):
63      risultato_da_arduino = risultato_da_arduino + " C'è parecchia luce."
64  if (valore_luce < 30):

```

Programma per Arduino:

- Sviluppato con l'IDE di Arduino
- Legge la seriale e intraprende le azioni in base ai comandi
- Lo sketch di Arduino va caricato mediante l'ambiente di sviluppo di Arduino

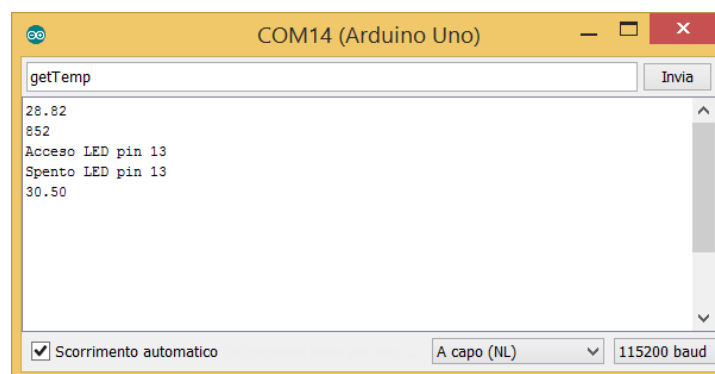


```

NAOduino
if (strcmp(command,"setLanguageENG")==0)
{
  Language='e';
  setLanguage=true;
}
if (strcmp(command,"getTemp")==0)
  getTemp=true;
if (strcmp(command,"getLight")==0)
  getLight=true;
if (strcmp(command,"setLed")==0)
  setLed=true;
if (strcmp(command,"resetLed")==0)
  resetLed=true;

if ( (getTemp==true) || (getLight==true) || (setLed==true) || (resetLed==true) || (setLanguage==true) )
{
  if (getTemp==true) acquisisciTemperaturaNTC();
  if (getLight==true) acquisisciLuce();
  if (setLed==true) {
    digitalWrite(13, HIGH);
    if (Language=='i')
      Serial.println("Acceso LED pin 13");
    else
      Serial.println("LED 13 spento");
  }
}
  
```

1 Arduino Uno on COM14



COM14 (Arduino Uno)

getTemp Invia

```

28.82
852
Acceso LED pin 13
Spento LED pin 13
30.50
  
```

Scorrimento automatico A capo (NL) 115200 baud

CONSIDERAZIONI FINALI

E' un progetto molto interessante e divertente, in cui ho avuto modo di scambiare idee e informazioni con varie persone.

Hardware:

L'idea iniziale era di collegare semplicemente una scheda Arduino.

Ho iniziato con lo starter kit di Arduino per poi passare a qualcosa che fosse il più essenziale e leggero possibile, da "fissare" sulla schiena di NAO con del velcro o con un elastico.

Al termine del progetto mi è venuta l'idea di realizzare uno "zainetto MeteoDuino", aggiungendo un sensore di umidità e trasformando NAO in un meteorologo del futuro ;-)



Software:

Per NAO ho sviluppato due versioni del software, in ITALIANO e in INGLESE. La versione in inglese all'inizio invia il messaggio "setLanguageENG", che istruisce Arduino a fornire feedback in lingua inglese.

Idee per ulteriori implementazioni software:

- far pubblicare a NAO su Internet i valori dei sensori meteo di Arduino
- far intraprendere a NAO azioni/dialoghi a seconda dei valori rilevati dai sensori di Arduino

Nota di colore: la prima versione di questo software l'ho messa a punto a Bologna presso la Fondazione Golinelli, durante il sabato sera e la domenica mattina della gara di robotica NAO Challenge 2015, con il sostegno e l'approvazione di colleghi, amici, visitatori e insegnanti.



Grazie per la vostra attenzione!



Pietro Alberti

pietro@mediadirect.it

01 Luglio 2015

Disclaimer: il sottoscritto declina ogni responsabilità per eventuali danni a persone e/o cose derivanti dall'uso del presente documento.