



UNIVERSITA' DEGLI STUDI DI MESSINA

DIPARTIMENTO DI MATEMATICA E INFORMATICA

Corso di Laurea in Informatica

L'uomo, i robot e l'intelligenza artificiale

Tesi di:
Luigi Chiricosta

Avvia

Anno accademico 2013/2014

Argomenti trattati

Storia

- dai telefoni agli smartphone

Software

- Brian e l'intelligenza artificiale

Hardware

- circuiti integrati

Conclusioni

- funzionamento finale

Obiettivi

- Brian, un software capace di simulare una conversazione e compiere determinate azioni (memorizzare appuntamenti, controllare le previsioni metereologiche fino a 15 giorni).

Obiettivi

- Bracciale, un dispositivo capace di scambiare messaggi audio con Brian tramite Bluetooth.

Obiettivi

- Ciabatta, un insieme di relè che comunicando con Brian tramite Ethernet realizzano un primo approccio ad un progetto di domotica (dal latino *domus*, casa, e *robotica*).

Storia

L'invenzione del telefono cellulare risale al 1973 ed è attribuita a Martin Cooper, direttore della sezione “Ricerca e Sviluppo” della Motorola.



Storia

Le prime comunicazioni avvenivano per via analogica ma con il progredire delle tecnologie ci si è diretti verso una comunicazione digitale, più veloce e sicura.

Storia

La comunicazione digitale ha attraversato diversi standard *Generation* (G) conosciuti con le seguenti sigle:

2G

3G

4G

Storia

Lo standard 2G è basato sull'uso del *GSM*, è uno standard aperto ed è il più diffuso al mondo e fu mantenuto dal consorzio *3GPP*.

Nasce il servizio degli SMS e quello della commutazione a pacchetto (grazie allo standard *GPRS/EDGE*) utilizzando il dispositivo come modem.

Storia

Il 3GPP fu creato per definire le specifiche tecniche dei dispositivi mobili della 3G che utilizzano lo standard *UMTS*.

Il canale di trasmissione audio *TDMA* viene soppiantato dal *W-CDMA* e questo comporta un notevole aumento della velocità di trasmissione dei dati e la nascita del servizio *MMS*.

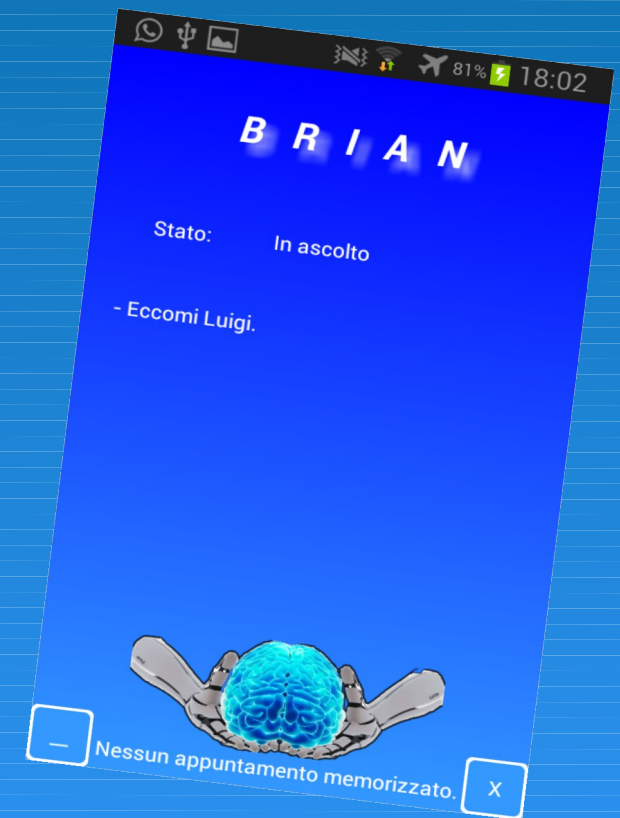
Storia

Dal 1973 al 1992 l'ingegneria elettronica si è evoluta in maniera così sorprendente che la miniaturizzazione dei circuiti integrati permette l'arrivo sul mercato di *Simon*, il primo cellulare intelligente capace di unire la mobilità di un cellulare alla capacità di calcolo di un computer. Era nato il primo *Smartphone*.

Storia

Ogni smartphone è gestito da un sistema operativo è il primo a diffondersi su larga scala fu il *Symbian*, utilizzato dai Nokia.

Al giorno d'oggi quelli più conosciuti e diffusi sono *Windows Phone*, *iOS* insieme a quello che è stato usato per la realizzazione di Brian, *Android*.



Software



Software

Il nome del progetto nasce dall'anagramma del termine inglese *Brain*.

L'idea è infatti quella di creare un software in grado di ascoltare e rispondere nel modo più simile possibile a come farebbe un qualsiasi essere umano, quasi come fosse dotato di un cervello.

Software

L'idea di realizzare una macchina intelligente risale al periodo del Nazismo durante il quale visse *Alan Turing*, matematico, logico e crittografo britannico, considerato uno dei più grandi matematici del XX secolo, nonché il padre dell'intelligenza artificiale, famoso per il *Test di Turing*.

Software

Il criterio del test, da lui precisato nell'articolo *Computing machinery and intelligence* della rivista *Mind*, prende spunto da *il gioco dell'imitazione* a tre partecipanti:

- un uomo, A
- una donna, B
- una terza persona, C

Software

C è tenuto separato da A e B e deve stabilire chi è l'uomo e chi la donna.

A deve ingannare C cercando di farlo sbagliare.

B deve aiutare C a risolvere correttamente il gioco.

Software

Il test si basa sul presupposto che una macchina si sostituisca ad A. Se la percentuale di volte in cui C distingue uomo e donna è simile prima e dopo la sostituzione di A con la macchina, questa deve essere definita intelligente.

Software

Il software di Brian è stato sviluppato su un sistema operativo con *Windows XP* e realizzato con l'IDE *Eclipse* per il sistema operativo *Android* ed è stato testato sulla versione 4.1.2 (*Jelly Bean*).

Software

L'applicazione è pensata per il totale controllo dello smartphone e si può vedere come strutturata in 4 parti:

- Home page
- Esempi
- Brian
- MiniBrian

Software

Il file *AndroidManifest.xml* contiene il seguente blocco di codice

```
<receiver android:enabled="true" android:name=".Starter"  
    android:permission="android.permission.RECEIVE_BOOT_COMPLETED">  
    <intent-filter>  
        <action android:name="android.intent.action.BOOT_COMPLETED" />  
        <category android:name="android.intent.category.DEFAULT" />  
    </intent-filter>  
</receiver>
```

che al verificarsi dell'evento *BOOT_COMPLETED* esegue *Starter.class*.

Software

Starter è un *BroadcastReceiver* che effettua un'operazione sola, avvia la home:

```
public class Starter extends BroadcastReceiver
{
    //Receiver rintracciato all'avvio del telefono che permette
    //di avviare automaticamente l'Activity di Brian
    @Override
    public void onReceive(Context context, Intent intent)
    {
        try{
            Intent i = new Intent(context, Brian.class);
            i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            context.startActivity(i);
        }catch(Exception e){
            Toast.makeText(context, "Impossibile avviare Brian.", Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }
    }
}
```

Software

E questa è

l'Activity

Brian.class!



UNIVERSITÀ DEGLI STUDI DI
MESSINA
DIPARTIMENTO DI MATEMATICA
E INFORMATICA
Corso di Laurea in Informatica

L'uomo, i robot e l'intelligenza
artificiale

Tesi di:

Luigi Chiricosta

Avvia

Anno accademico 2013/2014

Software

Cliccando sul
bottone con il
punto interrogativo
accediamo alla
sezione
Esempi.class.



Domande possibili

UNIVERSITÀ DEGLI STUDI DI
MESSINA
DIPARTIMENTO DI MATEMATICA
E INFORMATICA
Corso di Laurea in Informatica

L'uomo, i robot e l'intelligenza
artificiale

Tesi di:
Luigi Chiricosta

Anno accademico 2013/2014

?

Avvia

Software

Questa è una lista di alcune cose che Brian può fare.

Esempi di cose a cui posso rispondere

Portami in via ...

Dis/attiva modalità aereo.

Dis/attiva bluetooth.

Dis/attiva WiFi.

Dis/attiva connessione dati.

Dis/attiva GPS.

Dis/attiva flash.

Ricordami di ... tra ... minuti

Ricordami di ... alle

Software

Questa è una
lista di alcune
cose che Brian
può fare.

Esempi di cose a cui posso
rispondere

Conosci Siri?

Termina

Scatta foto

Brian

Qual è lo stato della batteria?

Grazie.

Cretino.

Rispondi (in chiamata).

Cambio lo stato del relè numer...

Software

Cliccando invece
su avvia
procediamo
all'avvio completo
di Brian.



UNIVERSITÀ DEGLI STUDI DI
MESSINA
DIPARTIMENTO DI MATEMATICA
E INFORMATICA

Corso di Laurea in Informatica

L'uomo, i robot e l'intelligenza
artificiale

Tesi di:

Luigi Chiricosta

Avvia

Anno accademico 2013/2014

Software

L'utente,
intuitivamente,
può capire cosa
Brian stia facendo
grazie alla sezione
apposita realizzata
con un oggetto
della classe
TextView.



Software

E quando
l'inizializzazione
è stata completata
Brian ci saluta!

B R I A N

Stato: In ascolto

- Eccomi Luigi.



Nessun appuntamento memorizzato.



Software

E infine ci facciamo
ricordare di
andare alla seduta
di laurea!

B R I A N

Stato: In ascolto

ricordami di andare alla seduta di laurea alle
10

- Eccomi Luigi.



—

Nessun appuntamento memorizzato.

X

Software

E lui ci terrà il
conto alla rovescia
fino alla nomina
di dottore in
informatica!

B R I A N

Stato: In ascolto

- andare alla seduta di laurea: memorizzato.
- Eccomi Luigi.



09:59:46

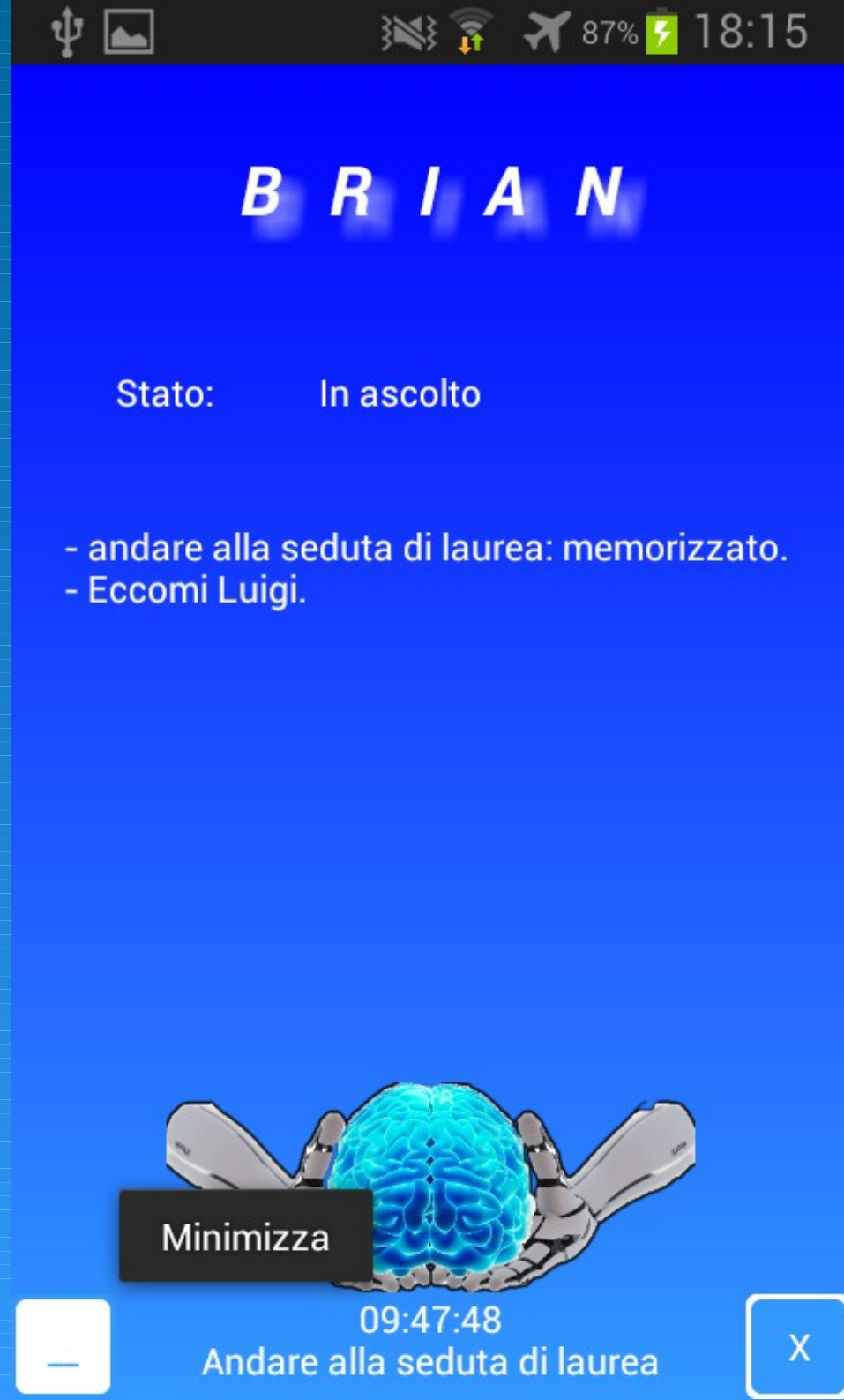
Andare alla seduta di laurea

—

X

Software

Per evitare di impegnare costantemente il display clicchiamo sul bottone *Minimizza*.



Software

E viene attivato
MiniBrian.class!

In ascolto

?



UNIVERSITÀ DEGLI STUDI DI
MESSINA
DIPARTIMENTO DI MATEMATICA
E INFORMATICA
Corso di Laurea in Informatica

L'uomo, i robot e l'intelligenza
artificiale

Tesi di:
Luigi Chiricosta

Torna

Anno accademico 2013/2014

Software

Brian si basa principalmente sull'uso di 2 librerie:

- RecognitionListner
- TextToSpeech

Software

RecognitionListner ci permette di registrare il suono in entrata dal microfono. Esso viene elaborato se risulta essere un suono di potenza media, ovvero sia compreso tra 40db e 60db.

Il suono viene poi elaborato e immagazzinato sotto forma di array di byte.

Software

TextToSpeech è una libreria che prende una stringa e dopo averla trasformata in array di byte la ritrasmette all'altoparlante sotto forma di onda sinusoidale.

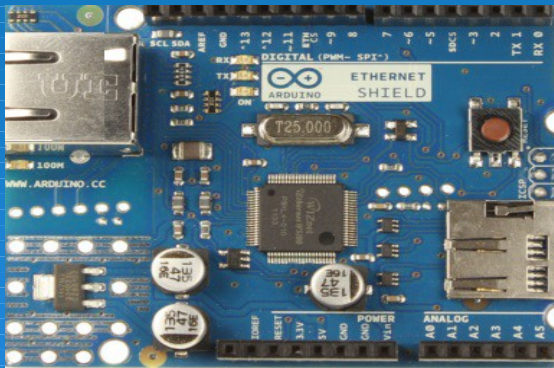
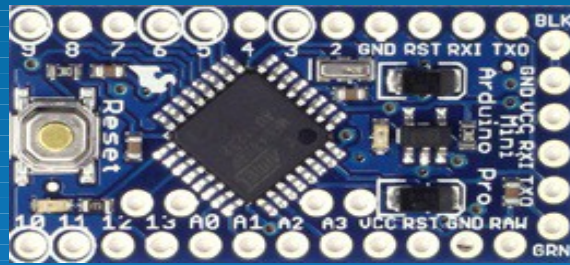
Software

In Brian, la classe che si occupa di gestire queste due librerie è *Dialogatore.class*.

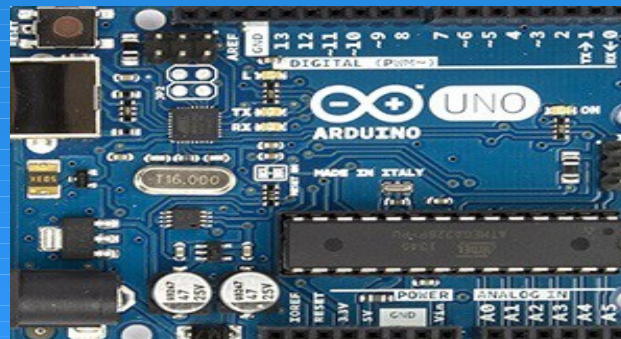
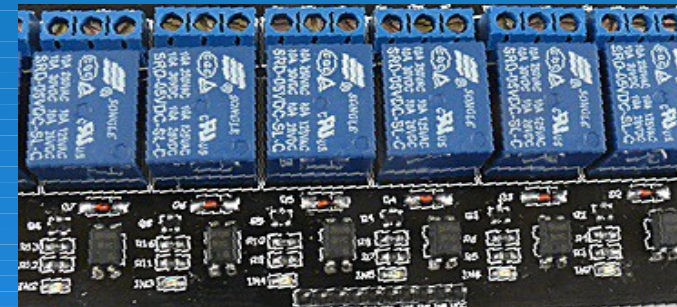
```
public class Dialogatore extends Service implements RecognitionListener, TextToSpeech.OnInitListener
{
    //variabile che tiene conto dello stato del service
    static boolean statoDialogatore;

    //variabili RecognitionListener
    String TAG = "RecognitionListener";
    static SpeechRecognizer sr;
    static Intent intentListener;
    protected static boolean fermaListen;
    private static int contatore;
    private int maxResults;

    //variabili TTS
    static TextToSpeech tts;
    private static String ttsMessage;
    private static boolean ttsInizializzato;
```

Hardware



Hardware

Alla base troviamo i *circuiti integrati*, componenti dalle dimensioni ridottissime dove i transistor sono formati tutti nello stesso istante tramite un processo fisico-chimico (crescita epitassiale).

Hardware

Gli integrati sono ampiamente utilizzati in elettronica e sono impiegati come microcontrollori per svolgere svariati compiti e realizzare unità aritmetico-logiche, aree di memoria, periferiche di I/O. E' possibile programmarli e ciò li rende versatili.

Hardware

In questo elaborato sono stati utilizzati microcontrollori della *Atmel* (utilizzati con *Arduino*) a cui verranno collegati diversi dispositivi come schede bluetooth, ethernet, auricolare e microfono.

Hardware

Il nome Arduino deriva da quello di un bar di Ivrea, frequentato da alcuni dei fondatori del progetto. Ivrea richiama a sua volta Arduino d'Ivrea, Re d'Italia nel 1002.

Hardware

Le *shield* di Arduino collegano, tramite circuito stampato, il microcontrollore ad alcuni pin connessi alle porte di I/O (analogiche e digitali), un regolatore di tensione, un'interfaccia USB e altro ancora in base a quella scelta.

Hardware

Per la realizzazione del bracciale si è pensato all'*Arduino Pro Mini* (*Atmega168*), date le sue dimensioni ridotte, mentre la ciabatta è stata realizzata con le shield *Arduino UNO* (*Atmega328*) poiché dotata di tantissimi pin di I/O.

Hardware

Le *breadboard* di Arduino sono programmabili tramite *Wiring*, linguaggio derivato dal C e dal C++, libero e quindi modificabile.

Hardware

La realizzazione del bracciale è stata pensata per una comunicazione più rapida e comoda con il dispositivo.

Dovendo infatti comunicarci via microfono, potrebbero essere scomodo dover utilizzare Brian sempre tenendo smartphone o tablet (alle volte molto grandi) in mano.

Hardware

Per costruire il bracciale abbiamo bisogno di:

- BreadBoard Bluetooth
- BreadBoard USB
- Microfono
- Auricolare
- LED
- Batteria

Hardware

La board USB ci serve per caricare il programma.

La batteria per poterlo alimentare senza fili.

La shield Bluetooth per scambiare messaggi tra software e hardware.

Il led per segnalare l'arrivo di byte tramite Bluetooth.

Il microfono per fare domande.

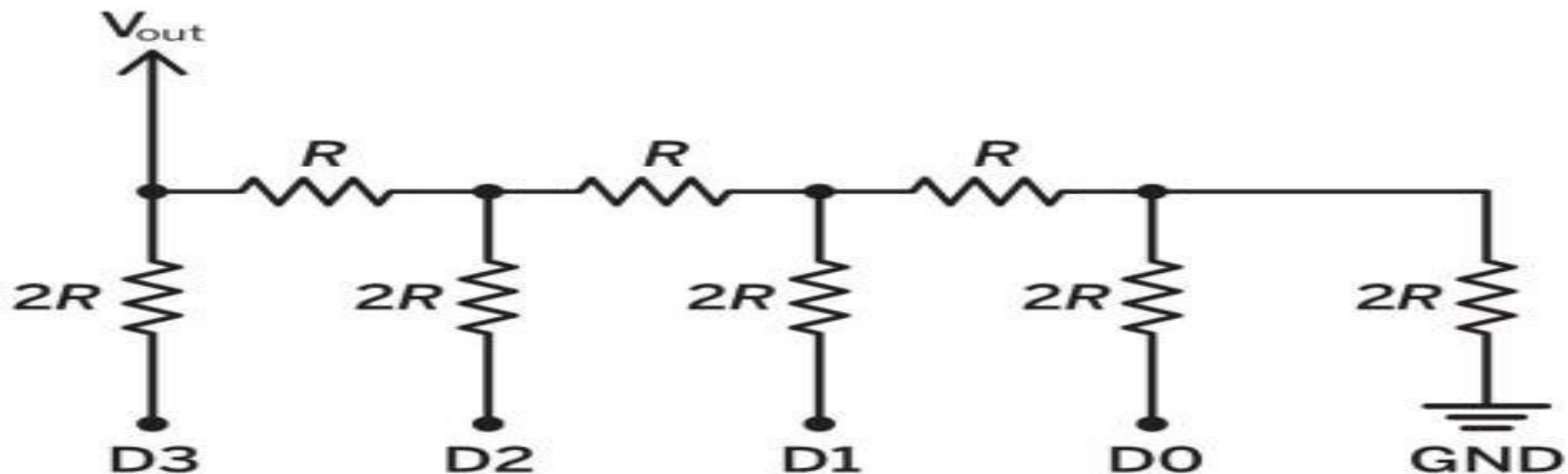
L'auricolare per sentire la risposta.

Hardware

L'unico problema è che questo microcontrollore è sprovvisto di pin analogici in uscita, quindi per ascoltare la risposta dovremo elaborare i byte per costruire un'onda sinusoidale tramite l'uso di un semplice circuito *DAC* a 4 ingressi.

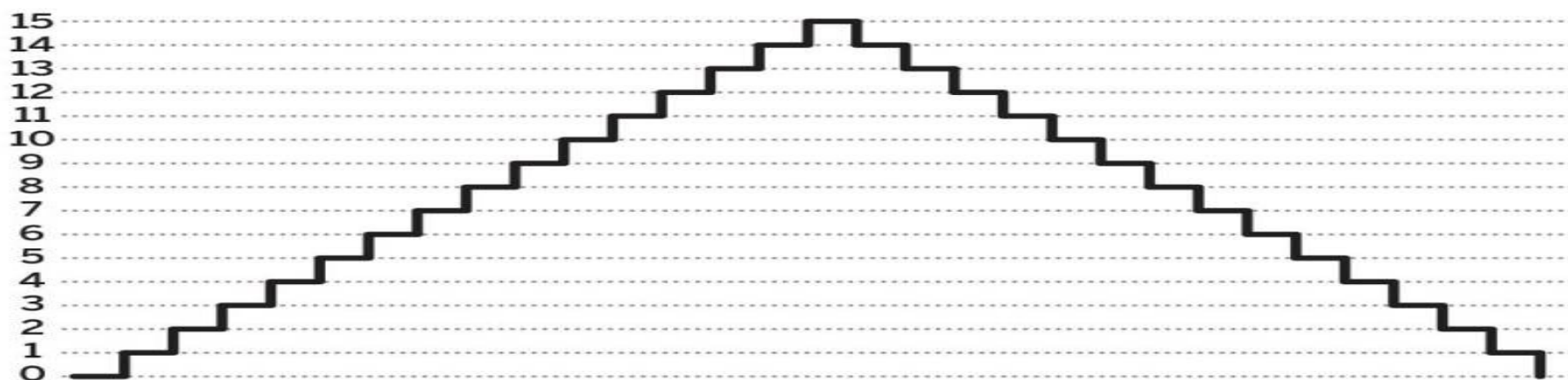
Hardware

Questo circuito trasforma i segnali in ingresso in una buona approssimazione di un'onda analogica.



Hardware

E questa immagine rappresenta l'onda ottenuta approssimando.



Hardware

La realizzazione della ciabatta è stata pensata per far sì che le capacità di Brian possano essere estese e così facendo possiamo far sì di comandare a distanza accensione e spegnimento dei più svariati oggetti realizzando un primo approccio alla *domotica*.

Hardware

Per costruire la ciabatta abbiamo bisogno di:

- Arduino Relè
- Arduino Ethernet Shield

Hardware

La shield dei relè sarà quella che effettivamente gestirà l'accendersi e lo spegnersi dei componenti a essi collegati.

La tensione massima supportata dai relè è di 230V in corrente alternata e di 30V in corrente continua, ma in entrambi i casi non più di 10A.

Hardware

E' costituita da 10 pin in ingresso.
I pin esterni sono quelli destinati alla VCC e al GND, mentre gli 8 interni servono ognuno alla gestione di un relè.

Hardware

La shield ethernet serve invece per poter gestire i relè a distanza idealmente infinita. Brian si occuperà di far variare dei valori inseriti in un database online e la shield dovrà collegarsi alla WAN e recuperarli periodicamente facendo eventualmente variare lo stato del relè corrispondente.



B R I A N

Stato: In ascolto

- Cambio stato relè 2 in corso.
- Eccomi Luigi.



TEST



Nessun appuntamento memorizzato.

X



B R I A N

Stato: In ascolto

- Cambio stato relè avvenuto.
- Cambio stato relè 2 in corso.
- Eccomi Luigi.



TEST



Nessun appuntamento memorizzato.

X

Funzionamento finale

Arduino Pro Mini, Arduino UNO e circa 10.000 righe di codice, funzionano così:

- Brian in attesa
- Ricezione
- Elaborazione
- Risposta

Funzionamento finale

Se il messaggio contiene la parola *relè* insieme ad un numero che lo identifica, Brian si collega al sito

<http://esameluigi.altervista.org/Brian/WebServer.php>

E ne cambia lo stato.

La shield ethernet, in continuo ascolto sul medesimo sito, quando trova una variazione di stato cambia il valore del relè.

Funzionamento finale

E se dicessimo “**Brian, accendi il relè 1**” e in quel relè è attaccato un fornellino con una caffettiera?

Bene, qualche minuto di pazienza e il caffè sarà realizzato dal nostro Brian!