

MAKING

# JARVIS

---

DISSENY D'INTERACCIÓ  
UNIVERSITAT OBERTA DE CATALUNYA

---

DAVID OTERO VERDAGUER

Bones Santi... bé, no sé què dir-te. Han estat 4 mesos per descobrir un entorn amb el que no havia treballat mai processing i Arduino; i anar donant pals de cec, superant poc a poc cada un dels reptes que suposava l'assistent.

Aquest document podria ser de moltes pàgines però no tinc ganes d'agobiar-te imaginant la quantitat de correccions que has de fer amb pocs dies per el que intentaré ser breu. Segur que em deixo coses però si alguna cosa no acaba de quedar prou ben explicada m'ho dius i et faré arribar una revisió.

A finals de gener d'aquest any, un company em va passar l'enllaç del que acabava de presentar Mark Zuckerberg. En aquest es podia veure al Mark a casa seu en plan IronMan!!! Quina passada, sóc un friki ho reconec!

Total que vaig pensar: Arduino, sensors, Linux... ostres perquè no ho provem?¿ i... aquí estem.

Com ja vares poder veure, la PAC va estar encaminada a veure com podia encendre i interactuar d'una manera "neta", ja que no volia fer molta instal·lació de relés, ni tenir cables per tota arreu. La idea de controlar-ho tot per radio freqüència va sortir llavors de veure que els infrarojos no eren la solució, com ja et vaig comentar. Un cop vaig tenir a Jarvis escoltant amb fluïdesa i amb poc marge d'error ja sols era qüestió de fer-lo parlar i pensar una mica.

## ENS ENDINSSEM A LA PRACTICA

### NODEJS I JOHNNY-FIVE

Si el descobriment de Processing va portar aquell joc dels ous i el pagès, el descobriment de NodeJS, del que encara en sóc completament novell, estic segur que m'ajudarà a fer virgueries. Node té una plataforma basada en JavaScript que treballa mitjançant Firmata, anomenada [Johnny-Five](#), que és una meravella.

Vaig arribar a ella posat que l'entorn de processing no m'acabava de fer el pes.

Vaig pensar que si es podia fer alguna cosa amb JavaScript seria molt millor. Així va ser com vaig trobar tot això.

La primera prova que vaig fer va ser lligar un termòmetre a un Arduino i veure com es comportava NodeJS. Vaig al·lucinar.

Node treballava de forma completament asíncrona i estava completament orientant a events. Això, en resum, volia dir que si creava un servidor web amb node i li passava la lectura del termòmetre podia veure el canvi de temperatura de forma continuada sense haver de refrescar pàgina ni utilitzar Ajax ni res per l'estil. Una autèntica passada!

Amb aquesta idea al cap i veient el seu funcionament vaig començar a fer proves: amb botons de JavaScript, amb leds, aplicant jQuery, bases de dades... cada cop al·lucinava més i més.

En poc temps vaig tenir un interfase web que encenia leds de l'arduino i és quan realment vaig pensar que la idea de Jarvis era possible.

## TCP SOCKETS

Molt bé, tenia 2 Arduinos. El primer, recollint informació del sensor de veu i emetent senyals de RF i IR. El segon, recollint informació de sensors de temperatura i lligat completament al servidor Debian; ja que NodeJS també podia executar comandes a mode de consola.

Pinta bé! Però? Com ho puc fer per comunicar un Arduino amb un altre?

La primera idea va ser fer-ho per RF ja que tenia les llibreries funcionant i sabia més o menys com fer-les servir. Però el problema va ser que s'havien d'enviar masses dades i d'una manera força fluida a poder ser amb comprovació de recepció, i és quan em va venir al cap el protocol TCP/IP.

Vaig començar a fer proves primer amb Arduino a veure si era capaç de rebre paquets TCP. Amb un programa lligat al meu PC, [PacketSender](#) feia les proves d'emetre paquets i Arduino els havia de reconèixer i interpretar.

No va ser molt complicat fer-lo funcionar. Ara tocava fer el mateix amb NodeJS, ja que la idea era que es parlessin entre ells confirmant la recepció. Rebaixaria el nivell a paquets UDP però faria una espècie de confirmació tipus TCP via missatges. Tot això ho pots veure funcionant en el vídeo quan en les consoles comenta **"he enviat/he rebut"**

Un incís, tot hagués estat més fàcil si ho hagués pogut fer tot amb un Arduino, però per una banda l'entorn directe de Arduino no m'ho permetia i per altra Johnny-Five no suportava el sensor de veu ni els emissor de RF i IR per el que forçadament havia de fer-ho així; crec! ;) A més d'aquesta forma podia fer que el mòdul de reconeixement de veu fos portable a qualsevol lloc de la casa ;)

## FESTIVAL

Ja tenia els Arduinos comunicant-se, Node escoltant events i treballant lligat directe al servidor Linux... ara tocava fer-lo parlar.

Amb tot això em vaig encaminar a software extern però les latències eren molt dolentes i no era fluid. Vaig recollir la idea de [festival](#) utilitzant un model de veu a baixa latència sacrificant qualitat d'àudio. En comptes de tirar amb veus Clunits, podia utilitzar un model HTS que era més ràpid per la lectura i més àgil a nivell de servidor. Funcionarà? ... Síiiiiii!

I em sona que festival té [veus en català](#), toma ya! La veritat que la llum del meu taller no s'apagava mai per els vespres... jajajaja

Fins les tantes de la matinada fent coses rares... aquest xicot!

Posat que com et comentava Node pot executar comandes de consola, estava segur ja que havia fet alguna cosa de logs i fitxers de text amb PHP i segur que Node podia millorar-ho, un cop l'Arduino 2 rep per socket o mitjançant un botó del servidor web una acció, es llença la comanda pertinent a festival per llegir el text programat. Tot amb molt poca latència.

## OPENCV

El primer que em va passar per el cap va ser reconèixer cares i prou, però veient la quantitat d'informació que hi havia a Internet i el fàcil que em va ser crear un programa per fer-ho amb el típic requadre verd, vaig dir-me... necessito més.

Així és quan vaig anar de web en web i de projecte en projecte per el GitHub fins trobar [OPENFACE](#).

He de dir que abans vaig trobar llibreries per executar OpenCV en Node però no acabava de rutllar.

OpenFace és força complex d'instal·lar en mode natiu. Per fer la prova pots clonar l'entorn de treball amb un sistema que es diu Docker per Linux, però no es viable treballar en aquest. Per el que un cop vaig veure que funcionava em vaig estar força temps per configurar un servidor nou amb les necessitats i parametritzacions de l'entorn de desenvolupament de OpenFace.

En aquest punt vaig haver de plantejar-me l'actualització del hardware del meu Debian. Tot plegat segur que no funcionaria en un P4 revell amb 1G de ram.

Vaig comprar una placa, una CPU, i ram i instal·lar-ho tot de nou. L'entorn LAMP que tenia i el que ara necessitaria per OpenFace.

El handicap més gran de OpenFace era que estava fet en Python i havia de modificar el codi per fer-lo funcionar enviant informació per sockets... total que ja tornem a estar liats amb un llenguatge del que no en tinc ni idea... jajajaja! Per cert, aquest la veritat que es força sensible amb els errors d'indexació i altres coses.

Openface demana que creïs una base de dades amb informació de les fotografies que mitjançant OpenCV detecta, transforma, retalla i qualifica. Un cop OpenCV ha fet la seva feina entra OpenFace creant una base de dades neuronal per tal de classificar la informació en matrius de punts. És un sistema una mica més avançat que el concepte de blobs, posat que no es tracta de fer un haar cascade per la detecció de figures, sinó d'anar més enllà.

La llibreria ha de ser capaç d'identificar matrius de punts, inclús en situacions una mica adverses i donar uns barems de coincidència, en funció d'aquests i ja dins el meu programa emetre una notificació i matar el programa de reconeixement per no gastar recursos.

I un dia tot va funcionar com calia! Quina passada, quina velocitat de reconèixer cares, quina fiabilitat! Això és la bomba ;)

## **ICLOUD**

Quan ja tenia tot el paquet més o menys funcionant vaig voler posar el toc al projecte. Un més ;)... Volia que el projecte fos funcional i m'ajudés en el dia a dia. Així que poder programar coses en agenda, i més tard accedir a funcions com la geolocalització donaria molt de suc a tot plegat.

Google i el seu calendar no em feia por, ja que aquesta gent comparteixen molt però Apple! Mare meva. Això és la cosa més tancada que existeix.

Després de fer mil proves amb CalDAV creant un propi client de calendaris Online vaig anar a parar a una altre llibreria de Python. [Pycloud](#).

Python ja no em feia por així que tot era qüestió d'hores i paciència...

### **... I AIXÒ ÉS TOT ;)**

Bé, Santi, no vull seguir perquè estic segur que estàs una mica esgotat de tanta paraulota en anglès i coses rares... només dir-te que el projecte no s'ha acabat i ara toca aplicar fasttext (intel·ligència en el us de paraules i aprenentatge) i alguna altra bajanada que em passi per el cap ;).

Em vaig posar un límit de 3 pàgines de text així que t'he de deixar... espero que t'hagi agradat el projecte i suposo que en els darrers dies obriré el projecte a tothom perquè ho puguin fer ells mateixos.

Per cert ara mateix el codi que t'adjunto és codi espagueti total. No està optimitzat, no és presentable, no hi ha funcions, hi ha errors... però el temps que he tingut per fer això més sol que la una ha estat limitat. Ara toca optimitzar.

Res més, un plaer. Una assignatura molt divertida i en la que he après un munt de coses!

Gràcies per tot Santi, i bon estiu!

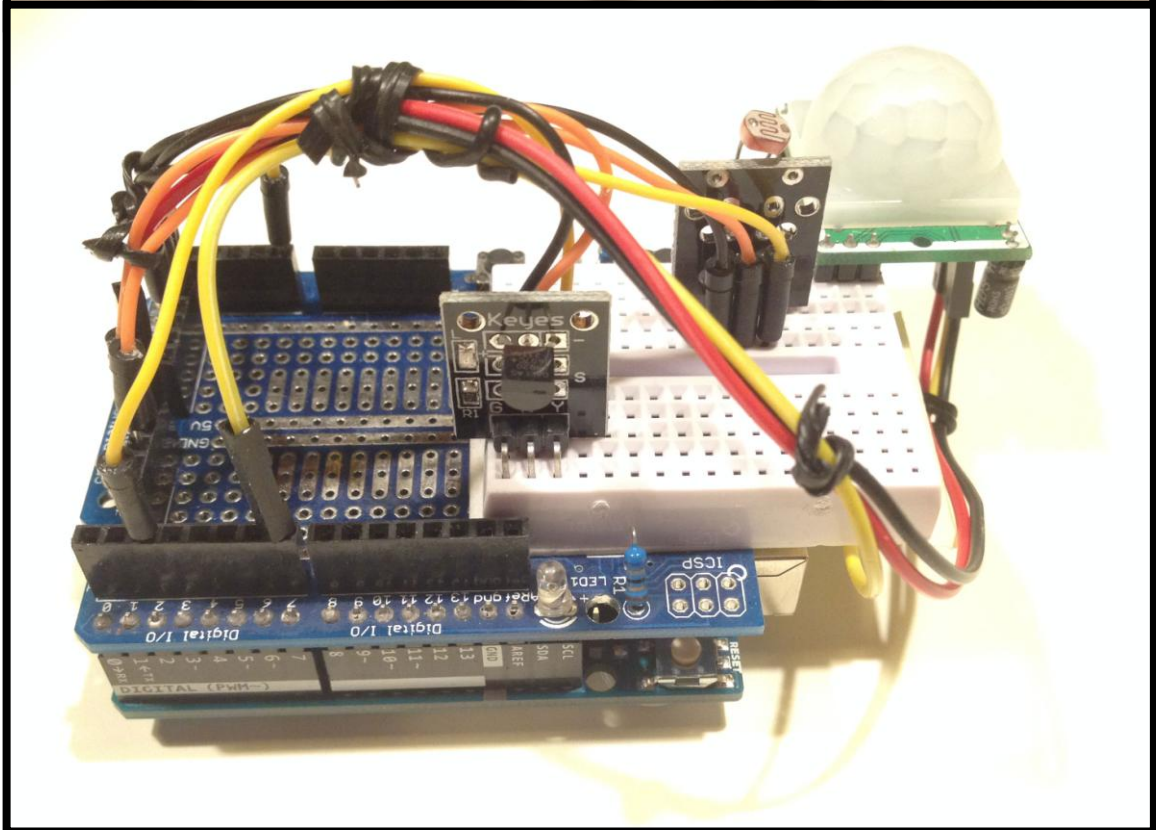
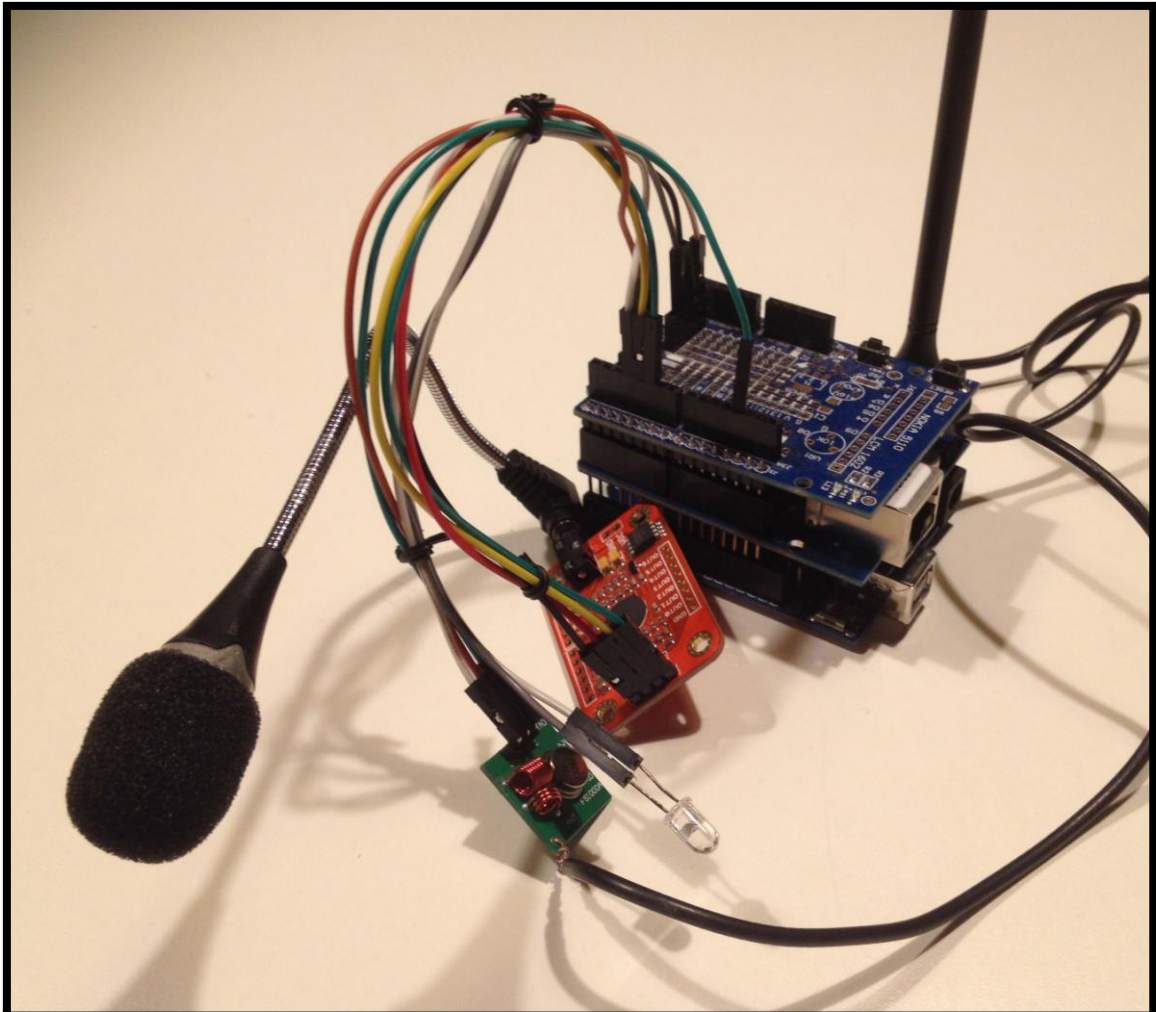
### **ENLLAÇ AL VIDEO EXPLICATIU:**

<https://vimeo.com/218971139> (Aquest és més per tots els públics però fes-li una ullada)

### **ENLLAÇ AL VIDEO D'ACCIÓ:**

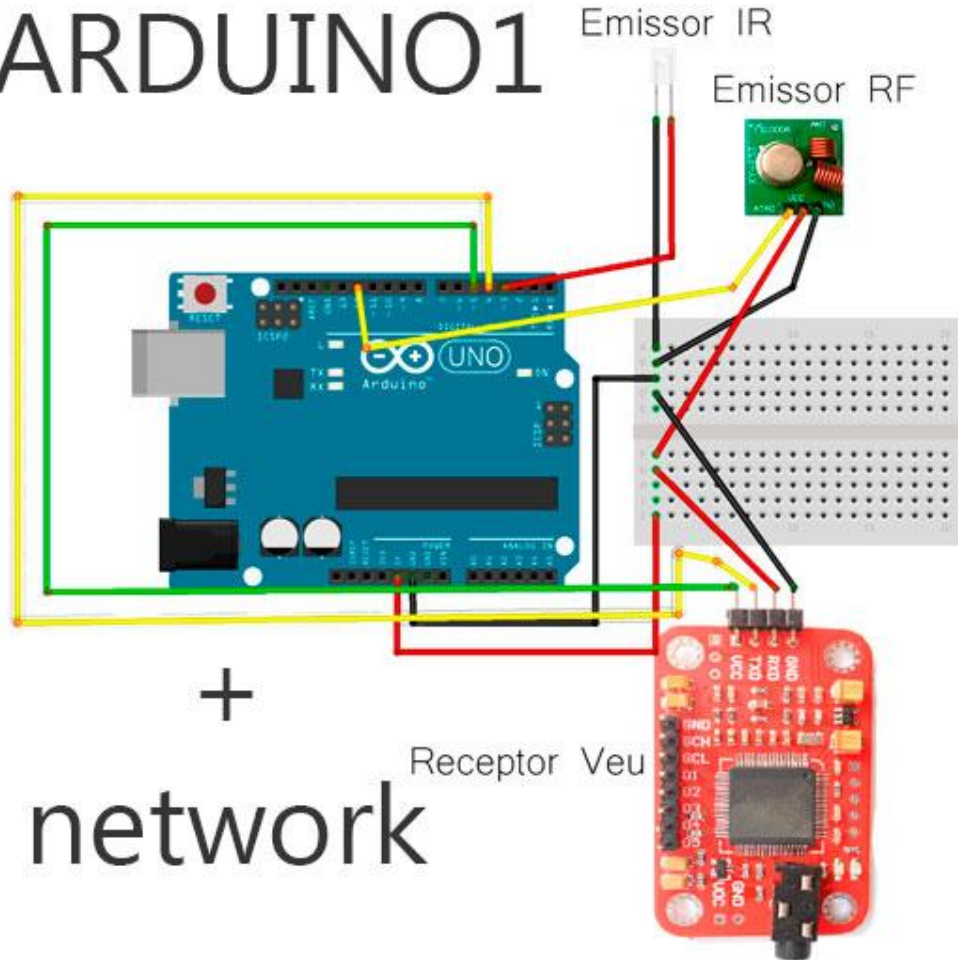
<https://vimeo.com/219125536>

Arduino 1 (control de veu i RF/IR) - Arduino 2(Sensor PIR/LLUM/TEMPERATURA)

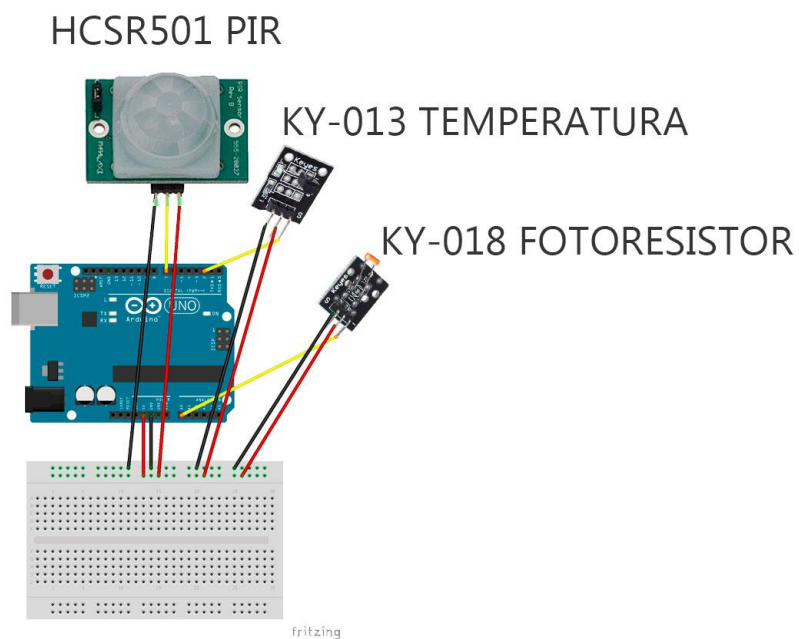


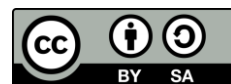


# ARDUINO1



# ARDUINO2





Disseny d'Interacció  
David Otero Verdaguer

UNIVERSITAT OBERTA DE CATALUNYA