



Treball de recerca

# Programació d'una maqueta automatitzada amb Arduino



Roger Camps Güell

2n Batxillerat A

Tutor: Maria Martínez

2016-2018

3/11/17

# Sumari:

1. Agraïments .....	1
2. Introducció.....	2
3. Que és la domòtica.....	3
4. La xarxa domòtica i les seves regulacions .....	5
5. Els sistemes domòtics i busos .....	6
• 5.1. Arquitectura Centralitzada:.....	6
• 5.2. Arquitectura Descentralitzada:.....	6
• 5.3. Arquitectura Distribuïda: .....	6
• 5.4. Arquitectura híbrida o mixta: .....	6
6. Els Elements Domòtics .....	7
<b>6.1. Actuadors:</b> Són aquells que fan les accions .....	7
6.1.1. Relé: .....	7
6.1.2. Motors Elèctrics: .....	7
6.1.3. Electrovàlvules:.....	8
6.1.4. Avisadors acústics i lluminosos:.....	8
<b>6.2. Sensors:</b> .....	8
6.2.1. De foc / fum: .....	8
6.2.4. D'hidrocarburs:.....	8
6.2.3. De monòxid de Carboni:.....	9
6.2.4. Detectores de presència:.....	9
7. Arduino.....	11
8. Placa Arduino Uno Rev 3.....	13
9. Placa de proves.....	15
11. Programació de la placa .....	18
11.1. Programació mitjançant Arduino IDE: .....	18
12. Programació de l'aplicació.....	21
13. Conclusions .....	24
14. Web grafia.....	25

# 1. Agraïments

En primer lloc vull agrair al meu tutor Maria Martínez tota la paciència i compromís que en aquests dies m'ha estat orientant i ajudant constantment per poder realitzar el treball adequadament.

En segon lloc vull agrair al meu antic tutor del treball de recerca de l'institut Castelló d'Empúries el tema proposat.

En tercer lloc però per mi la persona que m'ha guiat i m'ha proporcionat una quantitat d'informació i coneixement adients per realitzar el treball ha estat Adrià García Fundador de la pàgina web DIY makers.

Finalment, vull agrair a molt especialment el suport de totes les persones que han estat al costat durant la realització d'aquest treball.

## 2. Introducció

El tema triat me'l va proposar el meu antic tutor del treball de recerca, ja que en un començament jo volia fer un videojoc sandbox o free-roaming (món obert). El plantejament del treball era primerament fer un mapa en 3d utilitzant el programa *Unity 3d* per modelar les estructures en 3d i els personatges.

Però em va aconsellar que no el fes perquè requeria molt més temps del que disposàvem a més que gastaria molts diners en discs durs memòries usb, una tauleta gràfica. Així doncs em va proposar el tema de fer una maqueta d'una casa domòtica i em va semblar una bona idea, ja que englobava part de programació. Tanmateix no és un tema que em semblés interessant al començament. Encara que al llarg del treball m'ha agradat més.

Des d'un començament vaig escollir uns objectius clars i definits, com ara que havia de construir la maqueta, l'habitatge havia de poder ser controlat amb una app de telèfon intel·ligent feta a través d'app inventor o si més no programada perquè a unes certes hores fes una sèrie d'accions. I finalment aprendre a programar amb una placa Arduino. Juntament amb els objectius es plantejaven una sèrie d'hipòtesis com per exemple si era relativament senzill programar amb Arduino i quantes línies de codi es necessiten escriure perquè tot funcionés adequadament.

El treball està dividit clarament en dues parts, una la teòrica on s'expliquen diferents conceptes que es requeriran per poder comprendre la part pràctica.

### 3. Que és la domòtica

La paraula domòtica està formada per la paraula llatina domus (casa) i de la paraula grega tica (automàtica, que funciona per si sola). La domòtica és l'aplicació de les noves tecnologies als habitatges, que mitjançant sensors i el control de diferents dispositius electrònics és capaç de rebre informació processar-la i fer accions per tal de millorar la gestió energètica, la seguretat, el benestar i la comunicació dels seus habitants. Els sistemes domòtics normalment estan connectats a xarxes de comunicació o informació (Internet).

Actualment, és molt important saber diferenciar entre Immòtica i la Domòtica, la Immòtica és l'aplicació de les noves tecnologies a edificis que no són habitatges com ara magatzems, centres comercials, edificis d'oficines, etc.

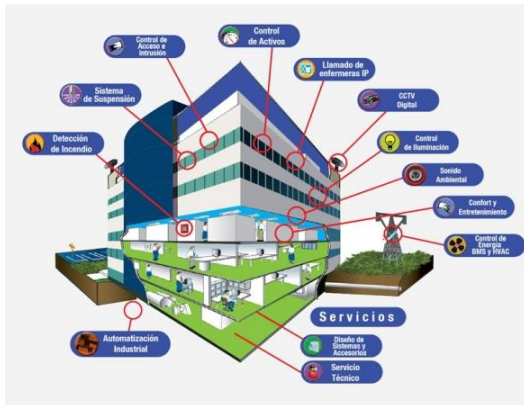


Figura 1 (Immòtica)



Figura 2 (Domòtica)

per altra banda la domòtica és l'aplicació de les noves tecnologies única i exclusivament als habitatges.

La domòtica durant els últims anys del segle XX i els primers del XXI ha experimentat un gran desenvolupament, millorant la qualitat de la instal·lació i el servei que ofereixen. Un cop instal·la una xarxa Domòtica, el servei de post venda garanteix un perfecte manteniment de tots els serveis.

Mitjançant els elements domòtics es poden controlar diferents elements com ara:

### **Panys elèctrics:**

No són gaire habituals en llars però sí en la immòtica per a edificis de grans empreses i blocs de pisos. Els panys elèctrics permeten l'obertura de la porta remotament. També són molt habituals els videoporters en els blocs de pisos que permeten a l'usuari veure qui vol entrar gràcies a una càmera instal·lada al portal del mateix edifici o intercomunicador.



Figura 3

### **Calefacció i Aire condicionat:**

La climatització de l'habitatge és un dels punts que més energia consumeixen i per tant dins de l'estalvi energètic s'ha de tenir molt en compte. Una bona manera d'estalviar energia és la combinació de la climatització amb la domòtica. Ja que podem programar el termòstat i els sensors que conformen la xarxa domòtica per activar-se i parar-se en unes certes temperatures. A més podem programar els motors i el climatitzador perquè a l'estiu s'obrin les persianes i no haver de connectar l'aire condicionat.



Figura 4

### **Làmpades i Il·luminària:**

La il·luminació controlada per un sistema Domòtic permet estalviar energia en connectar sensors de llum i sensors de presència. Ja que podem connectar el sistema perquè només encengui els llums quan detecta als habitants o en determinats moments del dia. A més també podem programar el sistema perquè durant la nit la làmpada no faci ús de tota la potència d'il·luminació i per tant no consumeixi tanta energia.

## 4. La xarxa domòtica i les seves regulacions

La xarxa Domòtica és un conjunt d'elements, ja siguin sensors, motors, electrodomèstics, sistemes de seguretat, etc. Els quals estan connectats entre si de manera que tot ells estan governats per un servidor el qual processa la informació rebuda a través d'Internet i acciona un seguit d'aparells que intenten millorar la qualitat de vida dels habitants de la llar.

La xarxa va agrupada amb les altres xarxes que solen estar presents en un habitatge (xarxa elèctrica, televisió i Internet), ja que és regulada pel reglament electrotècnic per a la baixa tensió (*REBT*<sup>1</sup>) aprovat l'any 2002<sup>2</sup>.



Figura 5: REBT

Les xarxes elèctriques a les quals estan connectades es divideixen en dos grups depenent del risc d'accident i dels tràmits que es requereixen per ser aprovades i instal·lades. El reglament electrotècnic de baixa tensió regula les garanties que han de complir les instal·lacions els quals rebin l'alimentació d'una font que subministri una tensió igual o inferior a 1000 volts si es tracta de corrent altern<sup>3</sup> o 1500 volts si es tracta de corrent continu<sup>4</sup>.

<sup>1</sup> <http://infonorma.gencat.cat/pdf/Instruccio1-2014.pdf> Actualització del REBT

<sup>2</sup> [https://www.boe.es/boe\\_catalan/dias/2002/10/01/pdfs/A02526-02720.pdf](https://www.boe.es/boe_catalan/dias/2002/10/01/pdfs/A02526-02720.pdf) REBT

<sup>3</sup> El corrent altern és un corrent elèctric que pot canviar al llarg del temps ja sigui en sentit o intensitat (té menys pèrdues que el corrent continu)

<sup>4</sup> El corrent continu va ser el primer que va ser descobert i es caracteritza per un flux constant de càrregues elèctriques cap a un únic sentit.

## 5. Els sistemes domòtics i busos

Els sistemes domòtics es divideixen en dos grans grups, els centralitzats i els no centralitzats d'aquests en surten variacions.

- 5.1. Arquitectura Centralitzada:

Pertanyen a aquest grup tots aquells sistemes que tenen una unitat central de processament intel·ligent. La unitat central rep informació dels diferents aparells i sensors i seguidament la processa. A continuació, la unitat central acciona diferents aparells en resposta a la informació rebuda.

- 5.2. Arquitectura Descentralitzada:

Pertanyen a aquest grup tots els sistemes que tenen més d'una unitat de processament. A cadascun dels controladors hi ha sensors i actuadors. L'avantatge que tenen les arquitectures descentralitzades és que redueixen la quantitat de cablejat que s'empra.

- 5.3. Arquitectura Distribuïda:

En aquesta arquitectura cada actuator és alhora sensor i controlador. Els controladors estan connectats entre ells per mitjà d'un bus<sup>5</sup> de dades i permeten així l'intercanvi d'informació.

- 5.4. Arquitectura híbrida o mixta:

Com el nom indica és una mescla de totes les anteriors, pot contenir més de dos controladors centrals, i també pot contenir actuadors amb els seus propis controladors els quals actuïn pel seu compte.

- 5.5. Busos:

**Cablejat propi:** utilitza un cable coaxial<sup>6</sup> o fibra òptica per connectar tots els dispositius

**Cablejat Compartit:** El sistema comparteix la informació mitjançant un cable que al mateix temps es fa servir per a altres serveis.

**Inalàmbric:** Els dispositius estan connectats mitjançant radiofreqüències o infrarojos

---

<sup>5</sup> Bus: És un sistema informàtic que transfereix dades entre components d'un ordinador o entre ordinadors. Està format per cables, circuit imprès, resistències, i altres circuits integrats.

<sup>6</sup> Cable Coaxial: És un cable utilitzat per transportar senyals elèctrics d'alta freqüència que té dos conductors concèntrics, un central, anomenat *viu*, encarregat de portar la informació, i un exterior, d'aspecte tubular, anomenat *malla* o *blindatge*, que serveix com a referència de terra i retorn dels corrents.



## 6. Els Elements Domòtics

Tots els sistemes domòtics contenen els elements descrits a continuació, l'única cosa que varia en funció dels sistemes i de l'espai que es pugui utilitzar a l'habitatge.

**6.1. Actuadors:** Són aquells que fan les accions.

### 6.1.1. Relé:

És un mecanisme elèctric que permet modificar l'estat d'un commutador accionant un electroimant. Està format per una bobina i uns contactes que el que fan és atreure un cos metàl·lic i tancar així el circuit elèctric. Una característica d'aquest element és que permet

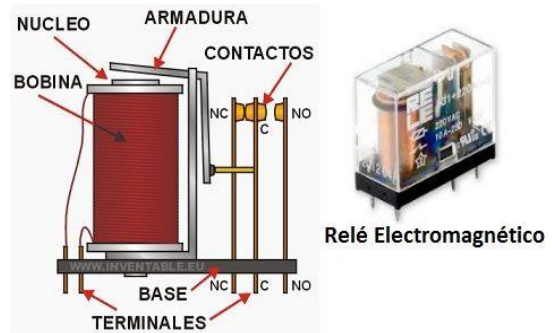


Figura 6: Relé

controlar circuits elèctrics amb intensitats i voltatges molt superiors al d'entrada. Hi ha tres tipus, el de blocatge, el de làmines i el de mercuri.

### 6.1.2. Motors Elèctrics:

Són uns elements que poden tenir un gran nombre d'aplicacions, com ara en persianes, portes de garatge... Tenen una gran importància dins la domàtica, ja que juntament amb els sensors de llum poden ser programats perquè s'accionin maximitzant l'eficiència energètica permetent l'entrada de llum i calor en funció de l'època de l'any en la que estem.

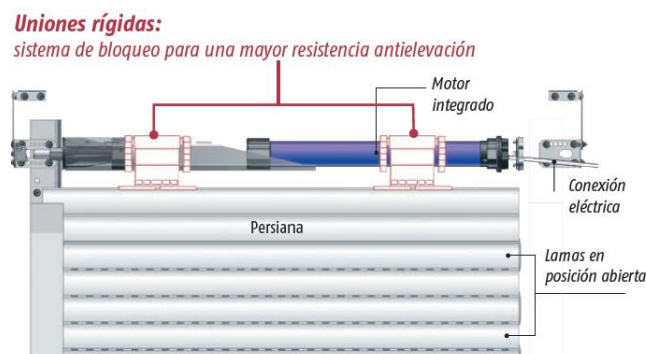


Figura 7: Motor

### 6.1.3. Electrovàlvules:

Permeten el control del subministrament d'aigua o gas a l'habitatge. Actualment s'instal·len a pràcticament tots els sistemes de reg i a les connexions que entren a l'habitatge. Ja que aquestes vàlvules consten de sensors que permeten saber si hi ha fugues i tancar la vàlvula per prevenir els possibles danys que es poguessin produir tant a la instal·lació com a l'habitatge.

### 6.1.4. Avisadors acústics i lluminosos:

Els avisadors normalment solen ser bronzidors i una llum d'emergència. Aquests solen estar connectats a diferents sensors com ara de moviment raigs infrarojos i de proximitat. Els avisadors se solen utilitzar com a alarmes per a incendis i per a intents de furts. A més solen estar connectats a la policia i a l'empresa que gestiona l'alarma.



Figura 8: Avisador acústic

## **6.2. Sensors:**

### 6.2.1. De foc / fum:

Se solen col·locar en punts estratègics on es poden produir flames o fum. Com ara a la cuina, on hi hagi la caldera, al quadre de fusibles.



Figura 9: Sensor de fum

### 6.2.4. D'hidrocarburs:

S'instal·len on puguin produir-se possibles fugues de gas, caldera, safareig, fogons. Hi ha un sensor per cada tipus de gas. Per tant si es canvia el tipus de gas, s'ha de canviar també el sensor. Per exemple si s'utilitza gas butà i es canvia per gas natural, s'ha de canviar també el sensor.



Figura 10: Sensor d'hidrocarburs

### 6.2.3. De monòxid de Carboni:

Aquest tipus de sensor s'instal·len en garatges i pàrquings subterranis o en llocs on hi hagi un gran moviment de vehicles que utilitzen com a combustible gasoil o gasolina.

El monòxid és un gas que es produeix en llocs on escasseja l'oxigen i no es pot formar diòxid de carboni.

És un gas molt tòxic insípid i inodor.



Figura11: Sensor de monòxid de Carboni

### 6.2.4. Detectores de presència:

Hi ha diferents detectors de presència, ja que cada un està basat en diferents tecnologies.

- **Detectors de presència basats en tecnologies de llum**

**infraroja:** Aquests detectors contenen un díode capaç de captar la llum infraroja que desprenen totes les coses tant éssers vius com objectes, i calibrar-lo perquè detecti com a un humà tot aquell cos capaç de produir una certa quantitat de llum infraroja.



Figura 12: Detector de presència

- **Detectors de presència basats en tecnologia de microones:** Són aquells que són capaços de detectar el mínim moviment d'un cos en el seu rang d'acció.

- **Detectors magnètics:** Es col·loquen en els marcs de les finestres i portes, aquests detecten la presència o no de l'ímant quan s'obren i fan saltar l'alarma.

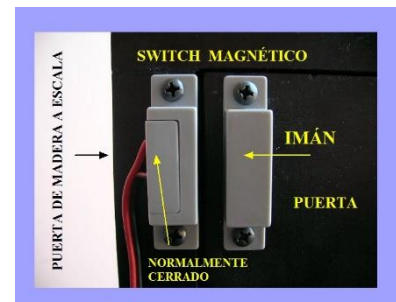


Figura 13: detector magnètic

- **Detectors de trencament de vidres:** Salta l'alarma si un vidre és trencat. Hi ha dos tipus, per contacte i acústics. Els de contacte es col·loquen directament sobre el vidre, mentre que els acústics quan detecten la freqüència del trencament d'un vidre, activen l'alarma.



Figura14: Detector de trencament de vidres

- **Barreres d'infrarojos:** Consten de dos elements, un emissor de rajos infrarojos i un receptor. Els corresponents elements es col·loquen alineats entre ells de manera que l'emissor emet rajos directament cap al receptor. En el moment en el qual el receptor deixa de rebre el raig, salta l'alarma. Es poden col·locar en diversos llocs, com ara portes, finestres, passadissos.

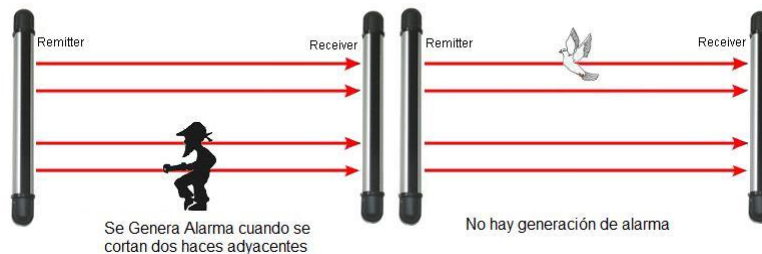


Figura15: Barreres d'infrarojos

- **Sensors de llum:** Detecten la quantitat de llum en luxs. Es poden instal·lar en les façanes exteriors, de manera que en funció del sol, es puguin controlar persianes, tendals.
- **Detectors de control d'accés:** Dins d'aquest grup s'inclouen tots aquells aparells com ara teclats numèrics, lectors, biomètrics de retina, lectors d'empremtes dactilars.



Figura16: Lector de retina

## 7. Arduino

Arduino és una plataforma de codi obert (open hardware), té hardware i software i està pensada perquè gent de totes les edats pugui utilitzar-la per poder desenvolupar projectes interactius.



El hardware d'Arduino conté molts elements però el principal són les plaques base. Arduino consta de més de 40 plaques però els models més venuts són: Arduino UNO rev 3, Arduino Leonardo, Arduino MEGA. Totes les Plaques Arduino consten d'un microcontrolador atmel, pins d'entrada i sortida, un jack per a l'alimentació i una entrada USB per a poder-la programar. Cada placa Arduino fa servir un controlador diferent, ATmega2560, ATmega 32u4, ATmega328.

A l'hora de programar les plaques, Arduino utilitza el codi C, és un codi senzill desenvolupat entre els anys 1969 i 1972 en els laboratoris de NokiaBell del qual deriven altres codis com

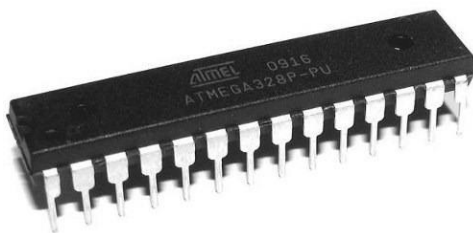


Figura 17: microcontrolador

ara el java, java script, C++, PHP entre molts d'altres. El codi C és l'evolució del codi de programació B i va suposar una estandardització en la manera que es programava.

Tot hi això el codi de llenguatge C té alguns inconvenients com ara que per a la gent principiant sigui més difícil de programar, perquè no afegeix ni corregeix parts de codi, es creen errors que amb altres tipus de llenguatge no passa, el compilador<sup>7</sup> del llenguatge C ha de reservar i alliberar memòria, dit d'una altra manera el compilador s'ha de dedicar de forma exclusiva a reservar la memòria per una part del codi mentre n'allibera. Això suposa una escriptura molt lenta i una gran càrrega tan humana com al maquinari.

Amb la plataforma Arduino pràcticament l'única limitació que es té, és la imaginació, ja que a la placa podem connectar sensors actuadors i leds a més de poder connectar multitud de mòduls com ara Bluetooth, ethernet, Gps, Wifi.

---

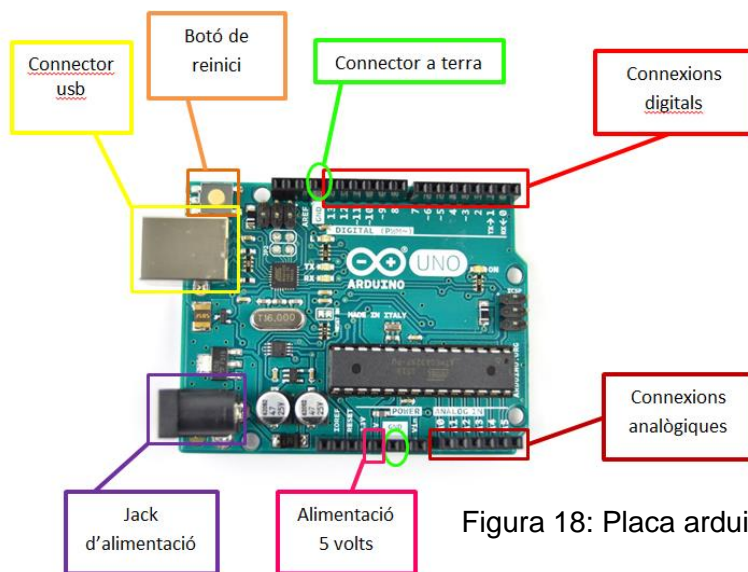
<sup>7</sup> Compilador: és un programa informàtic que tradueix un programa escrit en un llenguatge de programació a un altre llenguatge de programació, generant un programa equivalent que la màquina serà capaç d'interpretar.

# Part Pràctica



## 8. Placa Arduino Uno Rev 3

La placa Arduino Uno, fa servir l'ATmega328. Dins del microcontrolador hi ha una memòria flash de 32 kb dels quals, 0,5 els utilitza el bootloader<sup>8</sup>. El model Uno té 14 pins digitals que treballen a 5 volts, d'aquests 14, 6 tenen una sortida PWM<sup>9</sup> i proporcionen una sortida de 8 bits, cada un d'aquests poden donar i rebre un màxim de 40 mA. Alguns pins de les plaques Arduino tenen característiques especials. A aquest hi ha soldades diferents entrades amb diferents especificacions. A continuació hi han marcades les diferents parts i una explicació de cada una d'elles.



- **Connector USB:** el connector USB és on es connecta la placa Arduino amb l'ordinador, la placa es connecta a l'ordinador mitjançant un cable adaptador USB a tipus A. A més aquest cable permet alimentar el circuit mentre s'està programant.
- **Botó de reinici:** Tal com ens indica el seu nom permet reiniciar la placa mentre està en funcionament.
- **Connector a terra o GND:** en aquests pins és on es tanquen els circuits que es creen a la protoboard. El GND (ground en anglès), és el que fa que tot el voltatge que no absorbeixen els diferents aparells del circuit es converteixi a 0. Si el valor de GND no és de 0, se'n diu massa i és un valor baix.
- **Connexions digitals:** El model Uno conté 13 pins programables just a sota del botó de reinici. Aquests pins són on es connecten els actuadors que necessiten senyals

<sup>8</sup> Bootloader: és un gestor de arrancada, al encendre un aparell electrònic, és el primer que carrega per fer unes comprovacions i seleccionar quines parts de la memòria s'han de carregar les parts de la memòria

<sup>9</sup> PWM: és una tècnica emprada per a codificar missatges a través de polsos de un voltatge variable

digitals. A més hi ha alguns d'aquests pins que tenen característiques especials com per exemple poder variar la intensitat.



Figura 19: connexions digitals

- **Jack d'alimentació:** El jack d'alimentació és un connector circular situat a la superior i al costat contrari del botó de reinici. En aquest connector és on va connectada la font d'alimentació, és aconsellable alimentar la placa Arduino amb un voltatge d'uns 7,5 volts a 12 volts depenent del consum dels diferents components.

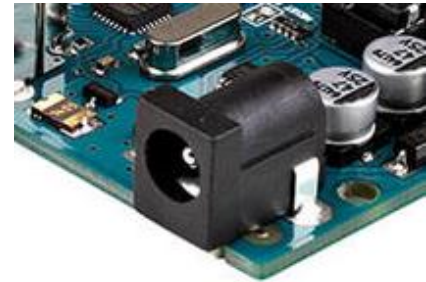


Figura 20: Jack d'alimentació

- **Pin d'alimentació 5 volts:** Aquest pin és per on surt el voltatge si algun dels components que estan connectats a la placa necessita una quantitat d'energia major i constant, aquest és el cas dels servomotors, els servomotors, necessiten una quantitat d'energia constant per poder realitzar les diferents accions, per això s'han d'alimentar amb aquest pin.



Figura 21: pin d'alimentació 5 volts

- **Connexions analògiques:** Els pins analògics és on es connecten els components que no necessiten un pin digital com per exemple els sensors. Estan situats a la banda contrària dels pins digitals. Els pins analògics van del A0 fins al A5 cada un d'aquests pins té 10 bits (1024 valors) i treballen a 5V alguns d'aquests pins analògics tenen característiques especials.



Figura 22: connexions analògiques



## 9. Placa de proves

Una placa de proves és una placa amb diferents orificis connectats entre si fets d'un material conductor per on pot circular l'electricitat. Normalment aquests orificis fets a la part superior de la placa segueixen uns patrons lineals. Està feta de dos materials un aïllant, normalment és plàstic o qualsevol polímer que sigui derivat de plàstics o aïllant. I el conductor, aquest és el que connecta les

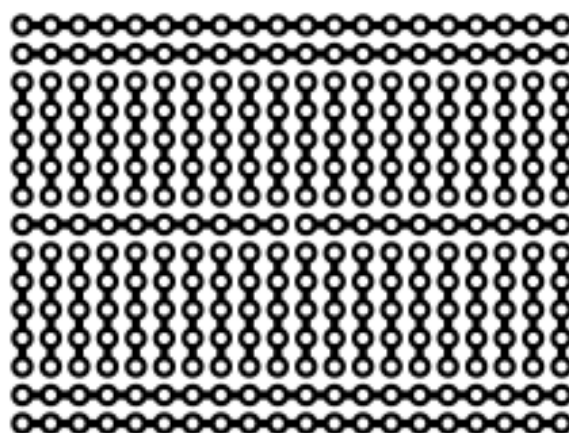


Figura 23: Patró d'una placa de

diferents seccions de la placa. En aquests orificis és on s'insereixen els diferents components. Les plaques de proves estan pensades per crear i comprovar circuits electrònics. Abans d'avançar cap a la producció industrial. De plaques, hi ha de diferents mesures i amb diferents nombres de pins. Però en general hi ha de dos tipus. Les d'ús permanent o temporal. Les plaques de proves d'ús permanent es diuen *Protoboard*<sup>10</sup> i les d'ús temporal *Stripboard*<sup>11</sup>

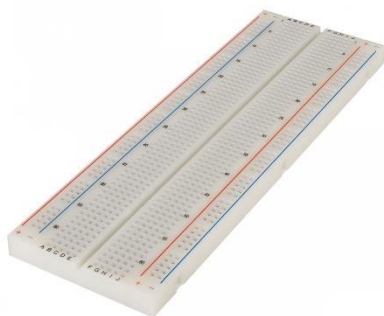


Figura 24: Protoboard

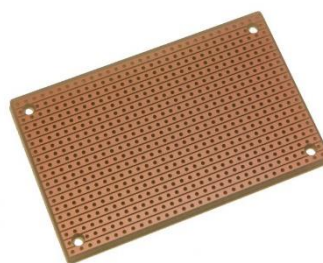

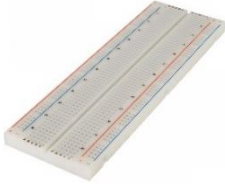
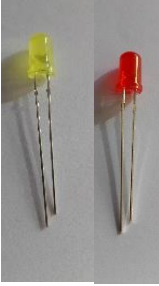






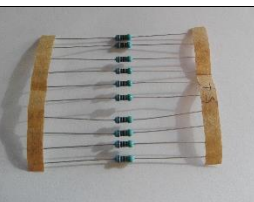
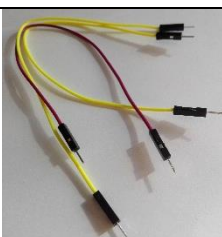

Figura 25: Stripboard

<sup>10</sup> És el tipus de placa més utilitzat, les línies laterals formen columnes connectades entre sí mentre que les centrals formen files horitzontals. Aquestes han de ser connectades mitjançant cables.

<sup>11</sup> Aquest tipus els diferents orificis no estan connectats entre si i els diferents components han de ser connectats mitjançant cables o soldadures.

## 10. Material utilitzat

Placa Arduino Uno REV 3	 A blue Arduino Uno REV 3 microcontroller board with a USB Type-B port, a DC power jack, and various integrated circuits and components.
Protoboard	 A standard white breadboard used for prototyping electronic circuits, featuring a grid of holes and two long power rails.
Led groc i vermell	 Two through-hole light-emitting diodes (LEDs), one yellow and one red, standing upright on a white surface.
Servomotor de rotació 360°	 A black 360-degree continuous rotation servo motor with a red cable and a black connector.
Brunzidor	 A black electrolytic capacitor with two metal leads extending from the bottom.
Polsador	 A yellow push button with a black base and two metal leads.

Mòdul Bluetooth	
Resistències	
Cable dupont mascle-masclle	
Cable dupont mascle-femella	

**Programes utilitzats:**

El programa escollit per a la programació de la placa ha estat Arduino IDE un software completament gratuït fet per la mateixa companyia. Per altra banda el programa escollit per a la creació de l'aplicació android ha estat app inventor del MIT que permet crear aplicacions des de zero.



Figura 25: programa app inventor

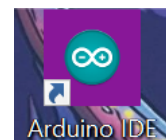


Figura 26: programa Arduino IDE

# 11. Programació de la placa

La maqueta consta d'il·luminació led, i motors per apujar i abaixar les persianes, a més d'un timbre que serà accionat amb un polsador. D'altra banda, la maqueta s'ha de poder controlar mitjançant una aplicació andorrid creada amb app inventor.

## 11.1. Programació mitjançant Arduino IDE:

En obrir el programa Arduino IDE s'obre una pestanya que es divideix en dues parts la void setup i la void loop. Per començar a programar, s'han de definir les variables. Això es fa a sobre de void setup utilitzant el #define. Primer s'ha d'introduir la llibreria de Bluetooth utilitzant SoftwareSerial BT(10 i 11) els números són els pins digitals on va connectat el mòdul Bluetooth. Seguidament, dins de void setup. S'ha d'escollir la velocitat amb la qual el mòdul pot transferir les dades així que es posa BT.begin (9600);



Figura 27: Void setup i Void loop

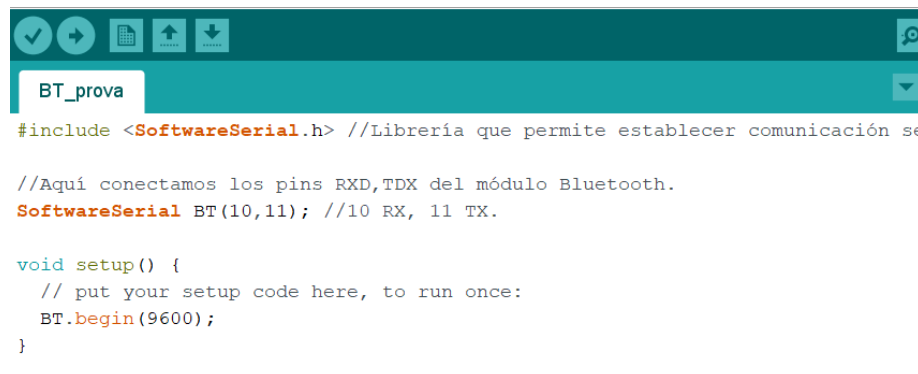


Figura 27: BT. begin

Per programar els servos, primerament s'ha d'introduir la llibreria com (servo.h) just a sota de la llibreria del Bluetooth. En segon lloc s'ha de decidir com es diran els leds i els ports digitals on estaran connectats. Seguidament, utilitzant (.attach) indiquem com es diuen els servos i a quins pins digitals van connectats. Aix3 perimetrà que a l'hora de programar el servo sigui m3s àgil. (mirar annexe pàgina 4)

Una vegada estiguin definides les sortides digitals, s'ha de crear el bucle que la placa estarà fent indefinidament mentre no l'hi carreguem cap altre programa.

La placa en tenir-se que controlar mitjançant una aplicació Android, s'ha d'utilitzar l'element (while) que significa mentre. Aquest element condicional farà que mentre la placa no detecti el valor escollit, no transmetrà la informació a cap actuator n'hi la transmetrà a cap led. A més dins d'aquest element, s'ha d'incloure l'element available que es tradueix com disponible. Aquest altre element farà que while s'activi només quan la connexió Bluetooth sigui possible.

```

while (BT.available()>0)
{

```

Figura 28: Element while i element available

Una vegada fet això es pot començar a programar els diferents elements.

- **Programació dels llums led:** Per a la programació d'un llum led he utilitzat, 7 cables dupont, un led, una resistència de 220 Homs, i una protoboard. Una vegada configurat el pin digital, dintre del void loop, configurem l'element (if) per quan rebi el caràcter en qüestió encengui o apagui el led: (if) (parametre=='a'). Perquè el pin pugui encendre el led, es posa dins del paràmetre (if) l'element digitalWrite (nom del led que es vulgui encendre, HIGH). Una vegada programat el led per encendre's s'ha de programar perquè s'apagui aleshores, obrim un altre (if) i l'hi donem un altre valor. A dins d'aquest segon (if) s'ha d'escriure digitalWrite (nom del led, LOW). Aquest procediment durant la programació s'ha repetit 7 vegades.

Mirar annexes pàgina 11

```

if(parametre == 'a')
{
    digitalWrite(ledG1, HIGH);
}
if(parametre == 'A')
{
    digitalWrite(ledG1, LOW);
}

```

Figura 29: programació per encendre i apagar leds

- **Programació dels servos:** per a la programació d'un servo es necessita 3 cables dupont ( voltatge, Gnd , Pin digital) i un servo una vegada activada la llibreria dels servos s'ha donat un nom al motor, s'ha de tenir en compte que: són servos que giren 360 graus. Per aquest motiu no es programen com servos normals sinó que la seva posició neutra és 90 graus. Per fer-los girar cap a la dreta, en comptes de posar: Nom del servo. write (90) es posen 180 graus i si vols que giri cap a l'esquerra has de posar 0 graus.

```

if(parametre == 'h')
{
servo.write(180); //Servo 1 gira en un sentit
delay(5000); //Girar durant 5 segons (canvia el temps si les persianes no a cavem de pujar)
servo.write(90); //Stop Servo 1
}

```

Figura 30: programació de servos

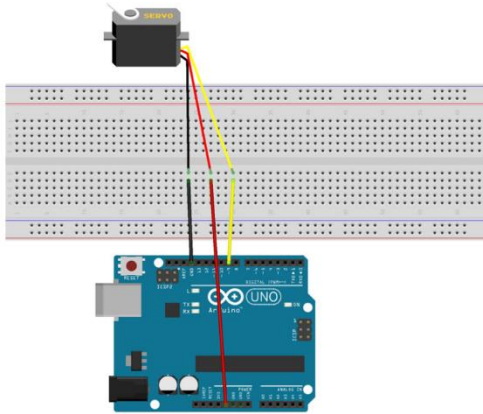


Figura 31: Connexió d'un servo

El delay és el temps que el servo girarà cap a la dreta i al cap de 5 segons pararà, com es pot veure a la figura 30. Mirar annexe pàgina si es vol connectar més d'un servo.

- **Programació del timbre /bronzidor:** Per programar el bronzidor, primerament,

```

#define pulsador 10
#define pinBuzzer 9

void setup() {
  // put your setup code here, to run once:
  pinMode(pulsador, INPUT);
}

void loop() {

  if(digitalRead(pulsador) == HIGH)
  {
    tone(pinBuzzer, 440);
  }
  if(digitalRead(pulsador) == LOW)
  {
    noTone(pinBuzzer);
  }
}

```

Figura: 32 programació d'un bronzidor

s'han de definir les sortides digitals en aquest cas la 9 pel bronzidor i la 10 pel pulsador, una vegada fet això, dintre el void setup, el pulsador ha d'estar identificat com a entrada per això l'anomenem INPUT. Perquè quan es premi el pulsador faci soroll, dintre de l'apartat de void loop, s'ha de col·locar el condicional (if) i dir, que si està polsat és un valor HIGH. Dintre d'aquest (if) s'ha

d'indicar a quina freqüència sonarà en aquest cas a 440 Hz. Per acabar, s'obre un altre condicional i es col·loca dins que si no està premut, el valor sigui LOW. Per dir que no soni dintre del segon (if) s'ha de posar noTone (pinBuzzer).

El bronzidor amaga un petit secret, aquest és que es poden fer melodies amb un bronzidor. Així que hi ha gent que ha creat, les melodies de cançons tan famoses com la de la marxa imperial de Star Wars, la de la introducció dels Simpsons, etc. Mirar annexe pàgina 6.

## 12. Programació de l'aplicació

Per controlar els diferents elements de la maqueta, era necessari crear una aplicació. El programa escollit va ser app inventor del MIT, ja que es basa en una programació per blocs molt senzilla, és gratuïta i permet aplicacions perquè es connectin per Bluetooth.



El programa té dues parts, una on es dissenya l'aspecte i els elements de l'aplicació i l'altre, l'apartat de programació on surten tots els elements que s'han utilitzat durant el disseny i les diferents accions que poden fer aquests en forma de blocs.

EN obrir un projecte demana el nom, en aquest cas TDR seguidament apareix una pantalla dividida en quatre parts, els elements que pots col·locar, la pantalla del dispositiu, com s'estructuren els diferents elements i les modificacions que es poden fer.

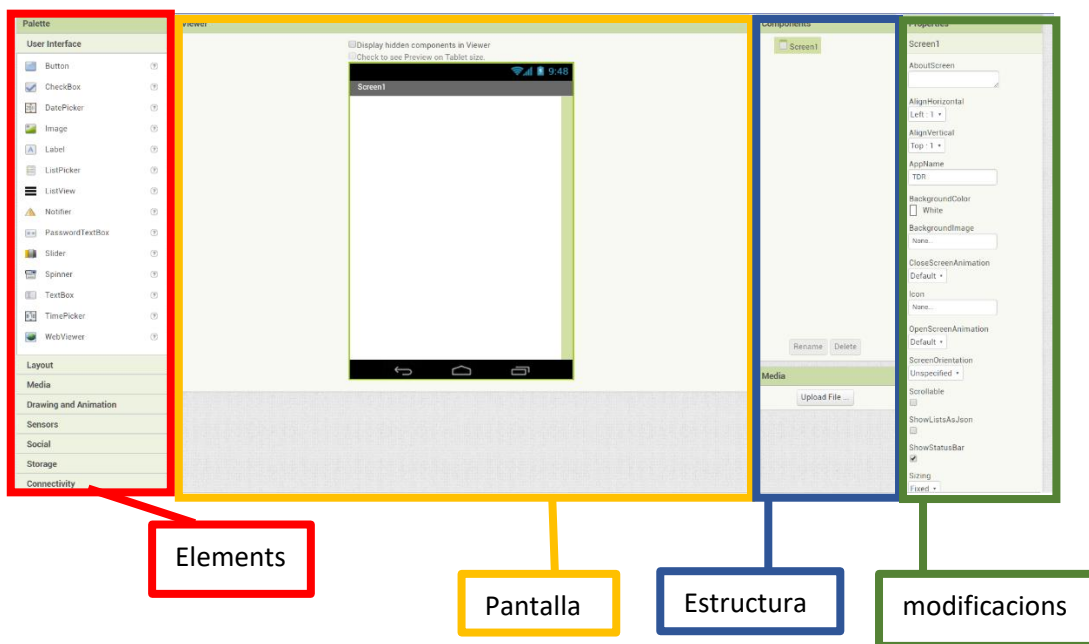


Figura 33: app inventor

L'aplicació que es requereix pel control d'Arduino, requereix botons, etiquetes, selectors de llista i un Bluetooth client.

El disseny de l'aplicació és diferent per cada persona, però el disseny realitzat en aquesta havia de ser la més intuïtiva possible per tant no feia falta distribuir els elements en diferents pantalles.



Figura 34: aplicació

En el meu cas l'aplicació consta d'un selector de llista que permet que s'obri una llista amb els diferents dispositius Bluetooth i connectar-s'hi. Tres taules amb diferents botons, a sobre de cada taula hi ha una etiqueta explicant quins elements controla. Dintre de cada taula, hi ha dos botons un amb la posició ON i l'altre amb la posició OFF.

La programació dels diferents elements ha estat relativament senzilla. Per programar el llistat d'aparells es necessita:

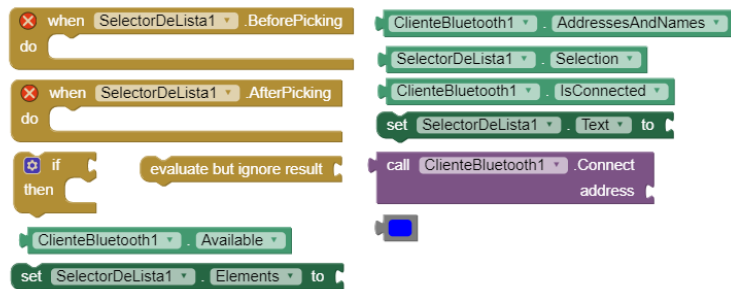


Figura 35: Blocs per a la programació

Els dos primers són dos condicionals que eviten que les accions es facin soles. El primer condicional s'activa si no el toques i el segon després de tocar-lo. El tercer element és un causal, el que fa és que si es compleix algun dels dos condicionals, automàticament fa la resta. Els de color verd són les accions que poden fer els dos elements en conjunt. Per unir el client Bluetooth i el Selector, es col·loca un al costat d'if i l'altre a sota.

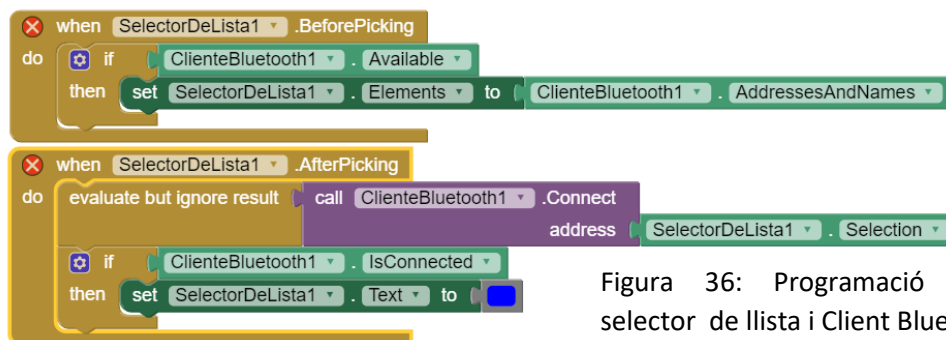


Figura 36: Programació dels elements selector de llista i Client Bluetooth

Al selector de la llista té un indicador perquè quan es connecti canviï de color el text. Seguidament s'han de programar els diferents botons perquè la placa Arduino a través de bluetooth pugui rebre els diferents caràcters per fer les accions.

Tal com hem fet amb el selector de llista, per programar els botons, necessitem un condicional un ( if then ), una acció del Bluetooth, i en aquest cas un caràcter que ha d'enviar.



Figura 37: Elements dels botons



La programació dels botons, el condicional amb dins el causal i dintre aquest hi ha una acció del Bluetooth i el que ha d'enviar.

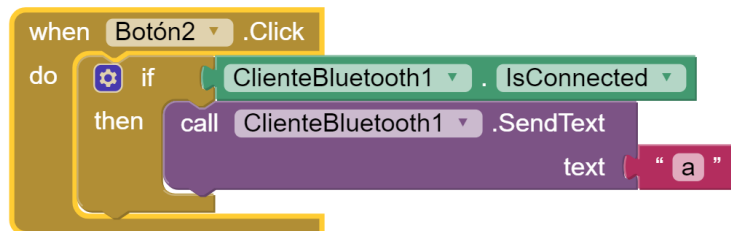


Figura 38: programació d'un botó

Tots els botons es programen de la mateixa manera i el que es fa, és duplicar un botó ja creat i canviar el botó que és i el caràcter que envia.

## 13. Conclusions

A l'inici del treball em vaig proposar alguns objectius, alguns més fàcils de realitzar que d'altres. Tot hi això, penso que he pogut superar les dificultats que anaven sortint al llarg del treball amb èxit.

Un dels objectius més fàcils d'aconseguir, era aprendre a programar amb el llenguatge d'Arduino (c). He hagut de consultar molta informació ja que pràcticament en sabia de programar. Tanmateix això no ha estat un impediment per poder acabar programant tot el codi com es veu en els annexos. Un dels problemes que més difícils de solucionar, ja que per mi suposava un dilema era la falta de pins de la placa, ja que només en té 13 i els pins 0 i 1 els utilitza a l'hora de carregar el problema a la memòria. Finalment, vaig solucionar aquest impàs comprant una altra placa Arduino. Per mi la part més difícil del treball ha estat aquesta, ja que la programació de l'aplicació al ser amb blocs, és molt més senzilla i intuïtiva. A més que ja s'havia programat amb App Inventor. Per tant, podria dir que l'objectiu d'aprendre a programar Arduino ha estat assolit amb èxit. Però, aquest no és l'únic objectiu que he pogut aconseguir el de dissenyar l'aplicació i controlar el sistema també han estat assolits amb èxit.

Una vegada acabat aquest treball, puc afirmar que programar Arduino és relativament senzill si ja saps programar, però si no la programació és molt feixuga i pesada, ja que hi has de dedicar un gran nombre d'hores.

Per acabar amb la conclusió del treball, hi havia un objectiu que era descobrir quantes línies de codi necessitava per poder programar el sistema domòtic. Aquest objectiu era curiositat meua i el nombre de línies necessàries són 130. Tanmateix no acaba aquí, A més es necessiten 82 línies de codi per programar la marxa imperial del brunzidor i 15 més per programar el polsador i enllaçar-lo amb el brunzidor.

En definitiva aquest treball tot i que al començament no m'entusiasmava gaire m'ha acabat agradant, m'he superat en l'àmbit de treball personal a l'hora de buscar les ajudes, les fonts i sobretot perquè ha estat un repte per mi.

## 14. Web grafia

<http://www.cedom.es/sobre-domotica/que-es-domotica> 2 setembre 2017

<http://www.monografias.com/trabajos93/domotica-seguridad-vivienda/domotica-seguridad-vivienda.shtml> 2 setembre 2017

[http://empresa.gencat.cat/ca/treb\\_ambits\\_actuacio/emo\\_energia\\_mines\\_seguretat\\_industrial/energia/energia\\_electrica/xarxes\\_de\\_distribucio\\_i\\_installacions\\_electriques/installacions\\_electriques\\_de\\_baixa\\_tensio/](http://empresa.gencat.cat/ca/treb_ambits_actuacio/emo_energia_mines_seguretat_industrial/energia/energia_electrica/xarxes_de_distribucio_i_installacions_electriques/installacions_electriques_de_baixa_tensio/) 4 setembre 2017

<https://www.definicionabc.com/tecnologia/bus-de-datos.php> 5 setembre 17

<https://www.prometec.net/> 20 setembre 2017

<http://diymakers.es/> 22 setembre 2017

[https://es.wikipedia.org/wiki/C\\_\(lenguaje\\_de\\_programaci%C3%B3n\)#Inconvenientes](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n)#Inconvenientes)  
22 setembre 2017

<https://definicion.de/compilador/> 25 setembre 2017

<https://www.androidpit.es/que-es-bootloader> 25 setembre 2017

[https://es.wikipedia.org/wiki/Placa\\_de\\_pruebas](https://es.wikipedia.org/wiki/Placa_de_pruebas) 27 setembre 2017

<https://programarfácil.com/tutoriales/fragmentos/servomotor-con-arduino/> 15 octubre 2017

[http://www.silvinopresa.com/wp-content/uploads/2015/08/4\\_servo\\_ard.png](http://www.silvinopresa.com/wp-content/uploads/2015/08/4_servo_ard.png)  
15 octubre 2017

<https://www.luisllamas.es/reproducir-sonidos-arduino-buzzer-pasivo-altavoz/>  
20 octubre 2017

<https://www.luisllamas.es/controlar-un-servo-de-rotacion-continua-con-arduino/>  
21 octubre 2017

<https://www.google.es/search?q=servo+rotacion+continua+arduino&og=servo+rotacion+c&aqs=chrome.1.69i57j0l5.5101j0j7&sourceid=chrome&ie=UTF-8> 21 octubre 17

<http://diymakers.es/crear-app-para-arduino-con-app-inventor/> 24 octubre 2017

<https://www.luisllamas.es/arduino-buzzer-activo/> 25 octubre 2017

# 15. Annexes

## **PROGRAMA COMPLET ARDUINO 1:**

```
k#include <SoftwareSerial.h>

//Librería que permet establir comunicació en sèrie en altres pins

#include <Servo.h>

#define ledV1 2
#define ledV2 3
#define ledV3 4
#define ledG1 5
#define ledG2 6
#define ledG3 7
#define ledG4 8
#define servo1_pin 11
#define servo2_pin 12
#define servo3_pin 13

//Crear els Servos
Servo servo1;
Servo servo2;
Servo servo3;

//Aquí connectem els pins RXD,TDX del mòdul Bluetooth.
SoftwareSerial BT(9,10); //9 RX, 10 TX.

void setup() {
  // put your setup code here, to run once:
  BT.begin(9600);

  pinMode(ledV1, OUTPUT); //significa que és un port de sortida, per aquí sortirà la
informació
  pinMode(ledV2, OUTPUT);
  pinMode(ledV3, OUTPUT);
  pinMode(ledG1, OUTPUT);
```

```
pinMode(ledG2, OUTPUT);
pinMode(ledG3, OUTPUT);
pinMode(ledG4, OUTPUT);
servo1.attach(servo1_pin);
servo2.attach(servo2_pin);
servo3.attach(servo3_pin)}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  while (BT.available()>0)
  {
    char parametre = BT.read();

    if(parametre == 'a')
    {
      digitalWrite(ledG1, HIGH);
    }
    if(parametre == 'A')
    {
      digitalWrite(ledG1, LOW);
    }
    if(parametre == 'b')
    {
      digitalWrite(ledG2, HIGH);
    }
    if(parametre == 'B')
    {
      digitalWrite(ledG2, LOW);
    }
    if(parametre == 'c')
    {
      digitalWrite(ledG3, HIGH);
```

```
}  
if(parametre == 'C')  
{  
    digitalWrite(ledG3, LOW);  
}  
if(parametre == 'd')  
{  
    digitalWrite(ledG4, HIGH);  
}  
if(parametre == 'D')  
{  
    digitalWrite(ledG3, LOW);  
}  
if(parametre == 'e')  
{  
    digitalWrite(ledV1, HIGH);  
}  
if(parametre == 'E')  
{  
    digitalWrite(ledG3, LOW);  
  
if(parametre == 'f')  
    {  
        digitalWrite(ledV2, HIGH);  
    }  
if(parametre == 'F')  
    {  
digitalWrite(ledG3, LOW);  
    }  
if(parametre == 'g')  
    {  
        digitalWrite(ledV3, HIGH);  
    }  
}
```

```

if(parametre == 'G')
{
  digitalWrite(ledG3, LOW);
}

if(parametre == 'h')
{
  servo1.write(180); //Servo 1 gira en un sentit
  delay(5000); //Girar durant 5 segons (canvia el temps si les persianes no a caven de pujar)
  servo1.write(90); //Stop Servo 1
}
if(parametre == 'i')
{
  servo1.write(0); //Servo 1 gira en l'altre sentit
  delay(5000); //Girar durant 5 segons (canvia el temps )
  servo1.write(90); //Stop Servo 1
}
if(parametre == 'j')
{
  servo2.write(180); //Servo 2 gira en un sentit
  delay(5000); //Girar durant 5 segons (canvia el temps quan ho ptovis a la maqueta)
  servo2.write(90); //Stop Servo 2
}
if(parametre == 'k')
{
  servo2.write(0); //Servo 2 gira en l'altre sentit
  delay(5000); //Girar durant 5 segons (canvia el temps quan ho ptovis a la maqueta)
  servo2.write(90); //Stop Servo 2
}
if(parametre == 'l')
{
  servo3.write(180); //Servo 3 gira en un sentit

```



```
    delay(5000); //Girar durant 5 segons (canvia el temps )
    servo3.write(90); //Stop Servo 3
  }
  if(parametre == 'm')
  {
    servo3.write(0); //Servo 3 gira en l'altre sentit
    delay(5000); //Girar durant 5 segons (canvia el temps )
    servo3.write(90); //Stop Servo 3
  }
}
}
```

### **Programació de el bronzidor:**

```
#define pulsador 10
#define pinBuzzer 9

void setup() {
  // put your setup code here, to run once:
  pinMode(pulsador, INPUT);
}

void loop() {

  if(digitalRead(pulsador) == HIGH)
  {
    tone(pinBuzzer, 440);
  }
  if(digitalRead(pulsador) == LOW)
  {
    noTone(pinBuzzer);
  }
}
```

### **Programació de la cançó del bronzidor:**

```
#define c3 7634
#define d3 6803
#define e3 6061
#define f3 5714
#define g3 5102
#define a3 4545
#define b3 4049
#define c4 3816 // 261 Hz
#define d4 3401 // 294 Hz
#define e4 3030 // 329 Hz
#define f4 2865 // 349 Hz
```

```

#define g4 2551 // 392 Hz
#define a4 2272 // 440 Hz
#define a4s 2146
#define b4 2028 // 493 Hz
#define c5 1912 // 523 Hz
#define d5 1706
#define d5s 1608
#define e5 1517 // 659 Hz
#define f5 1433 // 698 Hz
#define g5 1276
#define a5 1136
#define a5s 1073
#define b5 1012
#define c6 955

        #define R 0 // Define a special note, 'R', to represent a rest

// Melody 1: Star Wars Imperial March

int melody1[] = { a4, R, a4, R, a4, R, f4, R, c5, R, a4, R, f4, R, c5, R, a4, R, e5, R, e5,
R, e5, R, f5, R, c5, R, g5, R, f5, R, c5, R, a4, R};

int beats1[] = { 50, 20, 50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 80, 50, 20,
50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 40};

// Melody 2: Star Wars Theme

int melody2[] = { f4, f4, f4, a4s, f5, d5s, d5, c5, a5s, f5, d5s, d5, c5, a5s, f5, d5s, d5,
d5s, c5};

int beats2[] = { 21, 21, 21, 128, 128, 21, 21, 21, 128, 64, 21, 21, 21, 128, 64, 21,
21, 21, 128 };

int MAX_COUNT = sizeof(melody1) / 2;

long tempo = 10000;

int pause = 1000;

int rest_count = 50;

```

```

int toneM = 0;
int beat = 0;
long duration = 0;
int potVal = 0;
int speakerOut = 9;
int led = 6;

void setup() {

pinMode(speakerOut, OUTPUT);

pinMode(led, OUTPUT);
Serial.begin(9600);

}

void loop() {

// Melody1
for (int i=0; i<MAX_COUNT; i++) {
toneM = melody1[i];
beat = beats1[i];
duration = beat * tempo;
playTone();
delayMicroseconds(pause);
}
// Melody2
MAX_COUNT = sizeof(melody2) / 2;
for (int i = 0; i < MAX_COUNT; i++) {
toneM = melody2[i];
beat = beats2[i];
duration = beat * tempo;
}
}

```

```

playTone();
delayMicroseconds(pause);
}

}

void playTone() {

long elapsed_time = 0;
if (toneM > 0) {

    digitalWrite(led,HIGH);

while (elapsed_time < duration) {
digitalWrite(speakerOut,HIGH);
delayMicroseconds(toneM / 2);
digitalWrite(speakerOut, LOW);
delayMicroseconds(toneM / 2);
elapsed_time += (toneM);

    }

digitalWrite(led,LOW);

}
else {
for (int j = 0; j < rest_count; j++) {
delayMicroseconds(duration);
}

}
}

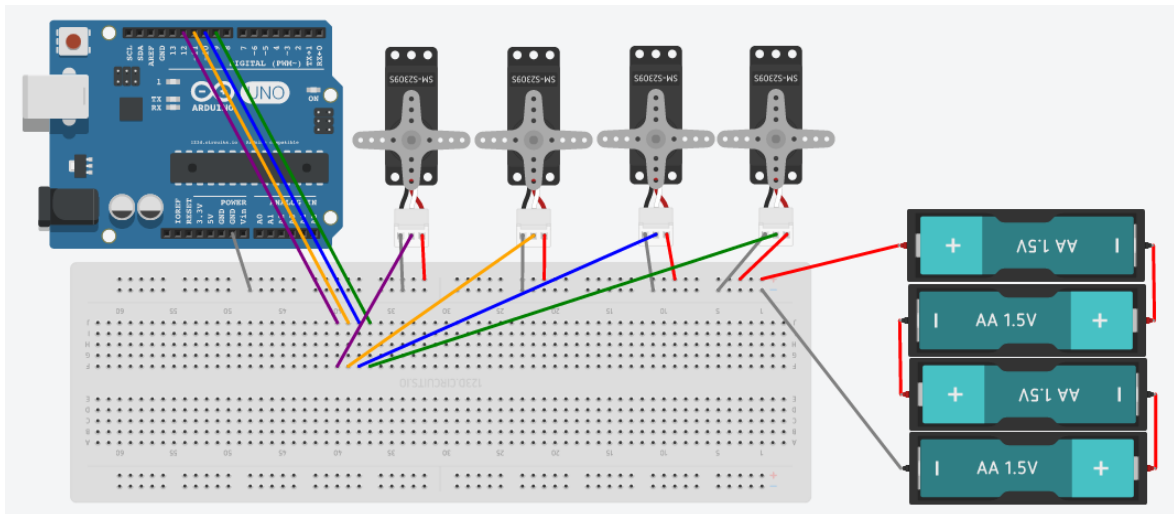
```

}

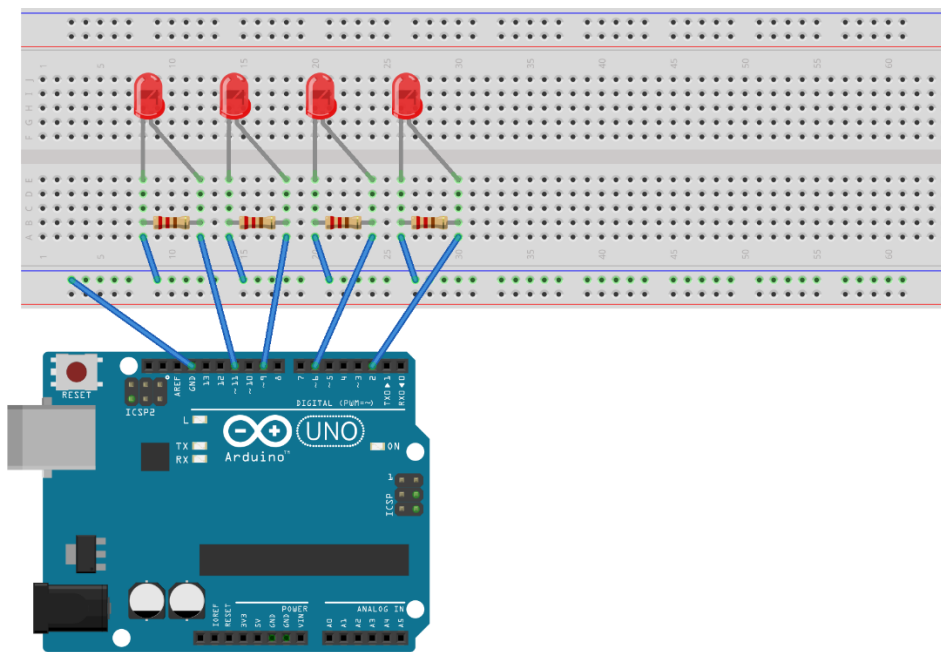
## Programació de l'aplicació



## Connexió de múltiples servos a una placa Arduino:



## Connexió de múltiples leds a una placa Arduino:



fritzing