

# L'HIVERNACLE INTEL·LIGENT



Laura Ysla Bueno

15 de gener del 2015

Tecnologia

Encarna García



*Vull agrair aquest treball a totes les persones que  
l'han fet possible, especialment als meus pares i a  
Encarna García*

INTRODUCCIÓ.....	3
1.1-QUÈ ÉS <i>ARDUINO</i> ? .....	5
1.-LA PLATAFORMA DE HARDWARE LIURE <i>ARDUINO</i> .....	5
1.2.-PLACA <i>ARDUINO</i> UNO .....	6
2.1-LLENGUATGE DE PROGRAMACIÓ DE <i>ARDUINO</i> : JAVA SCRIPT .....	9
2.-PROGRAMACIÓ DE L' <i>ARDUINO</i> .....	9
2.2-COS D'UN PROGRAMA AMB <i>ARDUINO</i> .....	12
3.- <i>APP INVENTOR</i> .....	15
4.1- TIPUS DE PLANTES .....	18
4.-DISSENY I CONTRUCCIÓ DE L'HIVERNACLE .....	18
4.2- DISSENY I MATERIALS .....	19
5.1-SISTEMES DE CONTROL .....	22
5.-DISSENY DEL SISTEMA DE CONTROL AUTOMÀTIC .....	22
5.1.1- TIPUS DE SISTEMA DE CONTROL .....	23
5.2- ESPECIFICACIONS DE DISSENY DE L'HIVERNACLE INTEL·LIGENT .....	24
5.3- ELEMENTS DEL NOSTRE SISTEMA DE CONTROL.....	25
5.3.1-ELEMENTS DE COMANDAMENTS MANUALS.....	26
5.3.2-SENSORS .....	26
5.3.3- CONTROLADORS.....	27
5.3.4- ACTUADORS .....	28
5.3.5-ELEMENTS DE COMUNICACIÓ: <i>BLUETOOTH</i> .....	28
6.-PROGRAMACIÓ DEL SISTEMA DE CONTROL AUTOMÀTIC .....	30
7.1- DISSENY DE LA PANTALLA .....	34
7.-APLICACIÓ: <i>SMART GREENHOUSE</i> .....	34
7.2-PROGRAMACIÓ DE L'APLICACIÓ .....	35
CONCLUSIONS .....	37
PROPOSTES DE MILLORA.....	39
BIBLIOGRAFIA .....	40
WEBGRAFIA.....	40
LLISTA DE REFERÈNCIES .....	40
ANNEX 1: PRESSUPOST DEL TREBALL .....	44
1.1-CÀLCUL DEL PREU HORA .....	44
1.1.1-COSTOS DIRECTES .....	44

# L'HIVERNACLE INTEL·LIGENT



1.-COSTOS D'ENGINYERIA.....	44
1.1.2-COSTOS INDIRECTES .....	45
1.1.3-PREU TOTAL.....	45
1.2-COSTOS DERIVATS DEL DISSENY .....	45
1.3-COSTOS DERIVATS DE L'ELABORACIÓ DE LA MEMÒRIA.....	46
1.4-COST TOTAL ENGINYERIA .....	46
2.1-CÀLCUL DEL PREU HORA .....	47
2.1.1-COSTOS DIRECTES .....	47
2.-COSTOS DERIVATS DEL MUNTATGE.....	47
2.1.2-COSTOS INDIRECTES .....	48
2.1.3-COSTOS TOTALS DERIVATS DEL MUNTATGE.....	48
2.2-COST TOTAL DEL MUNTATGE DE L'HIVERNACLE .....	48
ANNEX 2 : PROGRAMA .....	50





## INTRODUCCIÓ

Durant els últims anys el món ha evolucionat molt en aspectes relacionats amb la tecnologia i l'electrònica, ja que en molt poc temps hem passat a tenir Internet a casa, ordinadors, tablettes, *smartphones*, entre d'altres dispositius electrònics, que vint anys enrere hauria semblat impensable.

El meu interès per aquest àmbit es va iniciar quan vaig realitzar el treball de recerca de 4t d'ESO sobre automatització i domòtica. En concret vam realitzar una maqueta a escala del Pavelló Mies van der Rohe, transformat en habitatge i vam programar alguns sistemes automàtics en un microcontrolador *PICAXE*, controlat amb un comandament a distància.

Aquest projecte em va permetre veure a petita escala com es poden solucionar els problemes domèstics, i per aquest motiu em va semblar bona idea dur a terme un projecte relacionat amb el món de l'electrònica i l'automàtica.

D'altra banda, em vaig plantejar el repte de solucionar-li un problema als meus avis, ja que aquests comencen a estar grans i necessitem ajuda. No els hi agradava la idea d'haver de necessitar més ajuda per tal de mantenir vives les seves plantes, ja que aquestes els hi apassionen, i no volien haver de renunciar-hi. Per això, em vaig decidir a dissenyar i implementar un hivernacle intel·ligent que pogués proporcionar les condicions de temperatura i humitat necessàries pel creixement de les plantes. Tot això, supervisat per qualsevol familiar des d'un dispositiu mòbil *Android*.

Per implementar aquest projecte, en primer lloc havia d'escollir un microcontrolador que em permetés automatitzar tots els processos. Després d'estudiar entre les diferents possibilitats que ofereix el mercat (*PICAXE*, *PIC*, *Atmel*, *Arduino*, etc.) vaig decidir-me per la placa controladora *Arduino UNO*. Aquest tipus de plataforma de hardware és de codi obert, i em permet crear una interfície i afegir-li una gran gamma de dispositius per tal de fer-la més





complexa, per la qual cosa complia les condicions adients com a hardware. El principal motiu de la meua tria va ser que aquest microcontrolador em permetia dur a terme una idea més ambiciosa: connectar el meu hivernacle amb una aplicació mòbil.

L'*Arduino* UNO permet incorporar un mòdul de comunicacions *Bluetooth* que fa possible transmetre dades a una aplicació que podem instal·lar i utilitzar des de qualsevol dispositiu *Android* (tabletes, mòbil, etc.). Per tal de crear l'aplicació *Android* vaig utilitzar l'*App Inventor*, una plataforma *online* que ens permet crear aplicacions per a *Android* de forma senzilla i pràctica a l'hora d'utilitzar-la.

A l'hora de triar el tipus de plantes, vaig decidir utilitzar les que tenien els meus avis: plantes aromàtiques. Vaig utilitzar aquest tipus de plantes ja que anaven bé a l'hora de poder comparar la temperatura de l'interior i de l'exterior de l'hivernacle, ja que aquest té una part coberta i una altra descoberta, amb la qual cosa es podia fer un estudi complet d'aquestes i comparar els valors obtinguts.

Per assolir el meu objectiu de dissenyar, implementar i programar un hivernacle intel·ligent que donés resposta a les necessitats dels meus avis, he desenvolupat diferents parts en aquest treball de recerca. En primer lloc, es realitza una breu introducció als programes necessaris per programar el meu prototip. A continuació, es mostra com s'ha realitzat el disseny i la construcció de l'hivernacle i es descriuen els sistemes automàtics implementats. Finalment, s'explica el programa d'*Arduino* que controla els automatismes de l'hivernacle i l'aplicació *Android* que permet el control del prototip des d'un *Smartphone*.



## 1.-LA PLATAFORMA DE HARDWARE LIURE ARDUINO

El principal element d'un sistema de control automàtic és el microcontrolador. Actualment al mercat podem trobar multitud de marques i models que ofereixen diferents característiques i prestacions. Cada casa comercial fabrica models de microcontroladors que es programen emprant diferents llenguatges de programació.

Per escollir el microcontrolador pel prototip he tingut en compte aspectes com el cost econòmic, el llenguatge de programació que utilitza, el nombre d'entrades i sortides del mateix i la disponibilitat d'informació que hem permetés aprendre, de forma autodidacta, el seu funcionament.

Després d'analitzar tots aquest aspectes tècnics hem arribat a la conclusió que l'*Arduino* és el més adient. Els microcontroladors *ARDUINO* són dispositius de cost reduït (25 € aproximadament), són molt versàtils i es troben força documentats a Internet i a la literatura escrita. D'altra banda no requeria un nivell de programació gaire elevat alhora que t'introdueix al llenguatge de programació C.

### 1.1-QUÈ ÉS *ARDUINO*?

*Arduino* és una plataforma electrònica basada en *software* i *hardware* lliures, essent a la vegada flexibles i fàcils d'utilitzar. Està pensat per artistes, dissenyadors, aficionats i qualsevol interessat en crear entorns o objectes interactius.



Figura 1.1 Logotip d'*Arduino*





Aquesta plataforma electrònica pot rebre informació i dades del seu entorn a través dels seus pins d'entrada mitjançant tota la gamma de sensors que existeixen al mercat. Això fa que pugui ser utilitzada per controlar i actuar sobre tot allò que l'envolta: llums, motors i altres tipus d'actuadors. El microcontrolador<sup>1</sup> de la placa es programa utilitzant l'*Arduino Programming Language* basat en *Wiring*<sup>2</sup> i l'*Arduino Development Enviroment* basat en *Processing*<sup>3</sup>.

*Arduino* està compost per una àmplia gamma de plaques, que poden ser dissenyades i implementades per cada usuari o comprades directament de fàbrica, i el software per tal de programar-les pot ser descarregat de forma gratuïta.

### 1.2.-PLACA ARDUINO UNO

*Arduino* ofereix una àmplia gamma de productes, entre els quals estan les plaques, element necessari, ja que es tracta de la base dels programes. Aquestes són qui s'encarreguen de controlar l'entrada i relacionar-la amb la màquina o el procés que es vulgui dur a terme.

Aquesta plataforma electrònica ofereix als seus consumidors dinou tipus de plaques amb característiques diferents. De totes elles, vam triar la placa *Arduino UNO*, ja que aquesta era la que tenia les característiques necessàries per tal de poder automatitzar l'hivernacle, és a dir, no és gaire complexa per al meu treball, per la qual cosa es desaprofitaria ni molt bàsica cosa que no em permetria realitzar les meves propostes d'automatització.

---

<sup>1</sup> Un microcontrolador és un circuit integrat d'alta escala d'integració que incorpora la major part dels elements que configuren un controlador: procesador, memòria RAM, memòria per al programari, línies d'E/S, diversos mòduls pel control de perifèrics i generador d'impulsos de rellotge.

<sup>2</sup> És una plataforma de prototipació de programari obert compost per un llenguatge de programació, una interfície de desenvolupament integrada i una placa amb un microcontrolador.

<sup>3</sup> És un llenguatge de programació, un entorn de desenvolupament i una comunitat en línia. Consisteix en l'alfabetització de software dins de les arts i les cultures visuals dins la tecnologia.



## L'HIVERNACLE INTEL·LIGENT



Tot seguit explico més àmpliament les característiques d'aquesta placa per tal de poder, més endavant, entendre com es connecten els elements i com funcionen el programes.

La placa *Arduino UNO* és la placa més utilitzada pel seu caràcter d'aplicació general, ja que permet una programació assequible per a qualsevol persona amb nocions bàsiques de programació i el seu preu la fa molt assequible per educadors, dissenyadors i qualsevol interessat en electrònica i robòtica. Està composta per un microcontrolador *Atmega328* de l'empresa Atmel, té 14 pins digitals que poden utilitzar-se tant d'entrada com de sortida, funciona a 5 V i cada pin pot subministrar fins a 40 mA, essent aquesta quantitat també la intensitat màxima d'entrada.

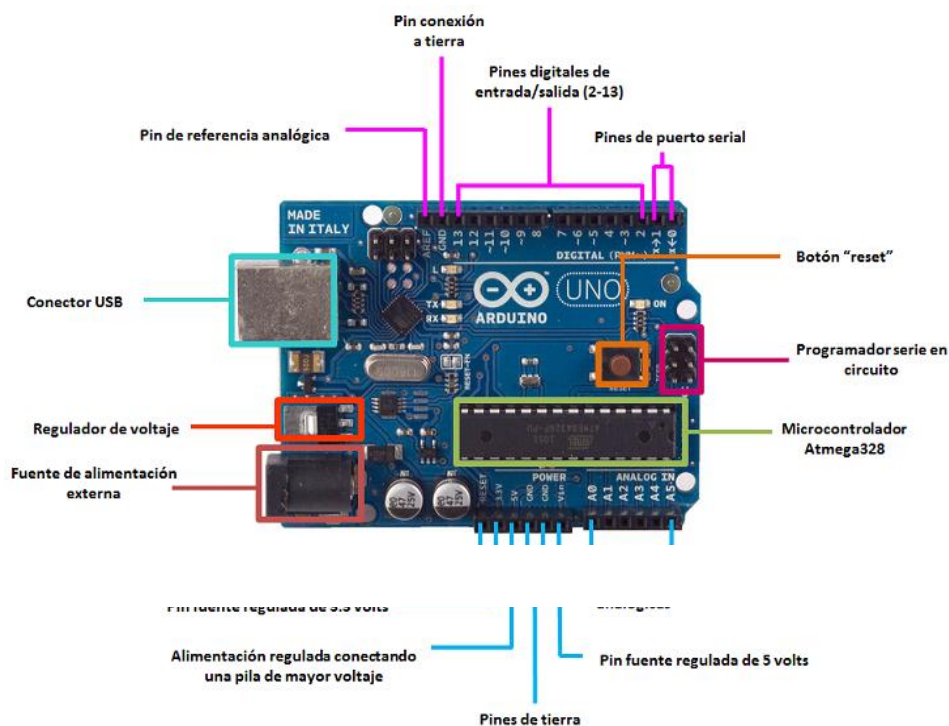


Figura 1.2 Parts de la placa *Arduino UNO*.

Com es resumeix a la taula de figura 1.2, també disposa de 6 pins d'entrada analògics que traslladen els senyals a un convertidor analògic/digital de 10 bits,





un 16 MHz ressonador ceràmic, connexió USB, un connector d'alimentació, un *header ICSP*<sup>4</sup> i un botó de reinicialització.

La placa pot alimentar-se directament mitjançant un cable USB o una font d'alimentació externa, com ara un petit transformador o una bateria de 5 V. És molt important no superar aquest voltatge per tal de no malmetre la placa.

Una de les característiques diferents d'aquesta placa és que utilitza cables estàndards, barats i fàcils de trobar. Aquesta no necessita en tot moment una bateria o un transformador, ja que mitjançant un USB se li pot subministrar el corrent elèctric necessari pel seu funcionament. La utilització de l'USB permet una ràpida transmissió de dades entre l'ordinador i la placa.

Característiques	Descripció
Microcontrolador	ATmega
Voltatge d'operació	5 V
Tensió d'entrada (recomanada)	7-12 V
Tensió d'entrada (límit)	6-20 V
Pins digitals E7S	14 (dels quals 6 proveeixen sortides PWM)
Pins d'entrada analògics	6
Corrent DC per pin E7S	40 mA
Corrent DC per pin 3.3 V	50 mA
Memòria Flash	16 KB (dels quals 2 KB són utilitzats per bootloader)
SPRAM	1 KB
EEPROM	512 bytes
Freqüència de rellotge	16 MHz

Figura 1.3 Taula resum de les principals característiques de la placa *Arduino UNO*.

<sup>4</sup> *In Circuit Serial Programming* (Programació Serial En Circuit) és un mètode de programació AVR, Parallax Propeller i PIC microcontrollers.



## 2.-PROGRAMACIÓ DE L'ARDUINO

### 2.1-LLENGUATGE DE PROGRAMACIÓ DE ARDUINO: JAVA SCRIPT

Un dels motius clau que m'ha resultat definitiu per escollir *Arduino* com a microcontrolador és el llenguatge de programació que utilitza. En aquest cas es tracta de Java Script, un llenguatge de programació semblant al C/C++ i per tant, orientat a objectes de programari lliure i de codi obert.

Un llenguatge de programació és un conjunt d'instruccions per a expressar operacions que poden ser dutes a terme per màquines com ara ordinadors o, com en el nostre cas, *Arduino*. Es pot utilitzar per crear programes que controlin el comportament físic i lògic d'una màquina, per expressar algoritmes amb precisió o com a mètode de comunicació humana.



Aquest llenguatge està format per una sèrie de símbols o instruccions i regles sintàctiques i semàntiques (estructures de programació) que defineixen una estructura i el significat dels seus elements i expressions. A la taula de la figura 2.2 es resumeixen les principals referències al llenguatge de programació d'*Arduino*.

Figura 2.1. Logotip de *Java Script*



Estructura de llenguatge	Funcionament
<i>void setup()</i>	S'utilitza per iniciar variables, establir l'estat dels pins, inicialitzar llibreries, etc.
<i>void loop()</i>	Executa consecutivament el codi assignat, permetent-li al programa variar i respondre.
<i>pinMode(pin, modo)</i>	Configura el pin especificat per comportar-se com una entrada (INPUT) o una sortida (OUTPUT).
<i>digitalWrite(pin, valor)</i>	Assigna el valor HIGH (5V) o LOW (0V) a un pin digital.
<i>digitalRead(pin)</i>	Llegeix el valor d'un pin digital especificat, HIGH o LOW.
<i>analogRead(pin)</i>	Llegeix el valor de tensió al pin analògic especificat.
<i>analogWrite(pin, valor)</i>	Escriu un valor analògic (PWM) en un pin. Pot ser utilitzat per controlar la lluminositat o la velocitat d'un motor.
<i>Serial.begin(speed)</i>	Estableix la velocitat de dades en bits/s per la transmissió de dades en sèrie.
<i>Serial.read()</i>	Llegeix les dades entrants del port sèrie.
<i>Serial.print(val, [format])</i>	Imprimeix les dades al port sèrie com a text ASCII <sup>5</sup>
<i>Serial.println(val, [format])</i>	Imprimeix les dades al port sèrie com a text ASCII seguit d'un retorn de carro i un caràcter d'avanç de línia.
<i>Serial.available()</i>	Retorna el número de bytes disponibles per a ser llegits pel port sèrie.
<i>If(condicional)</i>	Aquesta estructura comprova si certa condició es compleix.
<i>If/else</i>	Permet major control sobre el flux del codi que la declaració <i>if</i> , ja que permet agrupar múltiples comprovacions.

<sup>5</sup> ASCII és un codi de caràcters basats en l'alfabet llatí. Aquest codi utilitza 7 bits per representar els caràcters.



<i>If/else if</i>	Permet realitzar múltiples comprovacions en una mateixa estructura de condicions.
<i>Switch/case</i>	Controla el flux de programes permetent als programadors especificar diferents codis que haurien de ser executats en funció de varies condicions.
<i>while</i>	Els bucles <i>while</i> s'executen contínuament fins que la expressió del parèntesi passa a ser falsa.
<i>Do...while</i>	El bucle "do" treballa de la mateixa manera que el bucle <i>while</i> , amb l'excepció de que la condició es comprova al final del bucle, i això fa que el bucle s'executi "sempre" al menys una vegada.
<i>for</i>	S'utilitza per repetir un bloc tancat entre parèntesis. Aquesta estructura és molt útil per la majoria de les operacions repetitives.
<i>int</i>	Permet declarar les variables i definir un nom específic i també assignar un valor inicial.
<i>byte</i>	Emmagatzema un valor numèric de 8 bits sense punts decimals. Varia des de 0 fins a 255.
<i>array</i>	És una col·lecció de valors a la qual s'accedeix amb un índex numèric.

Figura 2.2 Taules de les principals estructures de llenguatge.



## 2.2-COS D'UN PROGRAMA AMB ARDUINO

El primer que hem de fer per poder crear un programa és saber què és l'IDE. IDE són les sigles d'Entorn Integrat de Desenvolupament. És una eina informàtica que serveix per al desenvolupament de programari de manera còmoda i ràpida, i agrupa les diferents funcions d'un sol programa: editor de codi, compilador, depurador i un programa de disseny.

Amb l'objectiu d'entendre l'estructura d'un programa qualsevol d'*Arduino*, estudiarem un exemple simple: encendre un *LED* que funcioni

de forma intermitent. L'algorisme d'aquest programa està representat al gràfic 2.3 en un diagrama de blocs. Com es pot veure, en primer lloc es necessari configurar una sèrie de paràmetres de l'*Arduino*; a continuació introduïm una instrucció per encendre un *LED*, afegim un temps d'espera i un cop fet aquest algorisme l'hem de passar a instruccions o sentències del llenguatge de programació Java Script, i per això hem d'elaborar una seqüència d'instruccions a través de les quals es reproduirà el procés definit anteriorment.

Per tal d'entendre l'estructura bàsica d'un programa, ja que aquesta ha de ser ordenada i estar ben definida, els programes es divideixen en tres parts: la capçalera i les variables i el cos del programa.

A la figura 2.4, encerclat en vermell, podem observar l'encapçalament del programa que estem realitzant. En aquesta part s'explica l'objectiu del programa i, si s'escau, alguns aclariments o especificacions del mateix. La seva funció és merament informativa i el text s'escriu entre els símbols " /\* " .

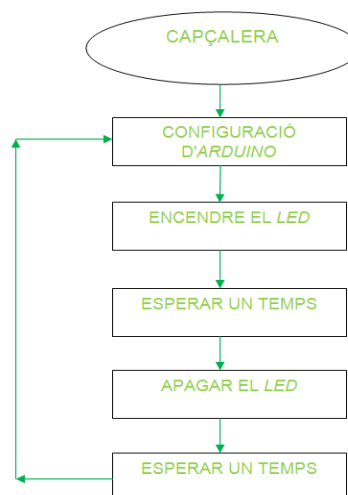


Figura 2.3 Disseny d'un algorisme gràfic.



Com podem observar al programa de la figura 2.4 encerclat en color lila, s'ha declarat la variable *LED* com a tipus (*int = led*). Es defineix que aquest *LED* es connecta a la entrada digital número 13.

A continuació, s'escriu el conjunt d'instruccions que constitueixen el cos del programa, que és el que podem veure encerclat en verd. Aquesta part té dues de

```

/*
  Parpelleig

  Encen i apaga un LED amb intervals d'1 segon,
*/

// El LED està connectat al Pin 13

int led = 13;

void setup(){
  pinMode(led,OUTPUT);
}

void loop(){
  digitalWrite(led,HIGH); //Encen el LED
  delay(1000);           // Espera 1 segon
  digitalWrite(led,LOW); // Apaga el LED
  delay(1000);           // Espera 1 segon
}

```

Figura 2.4 Programa: parpelleig d'un *LED*.

les estructures de llenguatges esmenades a la taula de la figura 2.2. La primera és la de *void setup()*, i s'encarrega d'iniciar la variable definida anteriorment. Dins d'aquesta funció trobem *pinMode*, i aquesta funció s'encarrega de configurar el *LED* com a una sortida. Quan tanquem la funció de *void setup()* trobem la funció de *void loop()*, la qual s'encarrega d'executar consecutivament el codi

assignat. Dins d'aquesta funció trobem dos funcions més: *digitalWrite*, que s'encarrega d'assignar el valor *HIGH*, que atorga els 5 V i *LOW*, que atorga els 0V al *LED* definit anteriorment; i *delay*, que s'encarrega d'establir el temps que el *LED* deixa de funcionar.

Com podem observar a les tres últimes línies, el final de cada instrucció està definit per un punt i coma. Aquesta forma de delimitar les instruccions és una norma bàsica del llenguatge de programació C. També podem observar que si escrivim dues barres ("*//*") podem aclarir el funcionament de l'ordre que estem utilitzant per tal que una altra persona pugui entendre en tot moment la funció que el nostre programa du a terme.

## L'HIVERNACLE INTEL·LIGENT



A més, cal tenir en compte que les instruccions que van entre claudàtors s'executen una vegada.

Finalment, hem de verificar si hi ha algun error al nostre programa o qualsevol instrucció escrita malament per tal de poder carregar el nostre programa a la placa *Arduino* i veure com funciona. Per carregar el programa a la placa només ens hem d'assegurar que el programa és correcte, connectar la placa amb l'ordinador i carregar el programa.





### 3.-APP INVENTOR

*App Inventor* és un entorn integrat de desenvolupament que permet crear aplicacions mòbils per al sistema operatiu *Android*. Aquest sistema per crear aplicacions és gratuït i permet crear d'una manera senzilla diferents tipus d'aplicacions.



Figura 3.1 Logotip d'*App Inventor*.

Per crear qualsevol aplicació el primer pas a dur a terme és realitzar el disseny gràfic de la pantalla, és a dir, fer un disseny que inclogui tots els elements necessaris per controlar la nostra aplicació (botons, imatges, text de contrasenyes, quadres de text, *Twitter*, etc.) i que aquesta resulti estètica i intuïtiva, és a dir, que no sigui difícil d'utilitzar i que tot estigui on ha d'estar per tal

de fer-la clara i precisa.

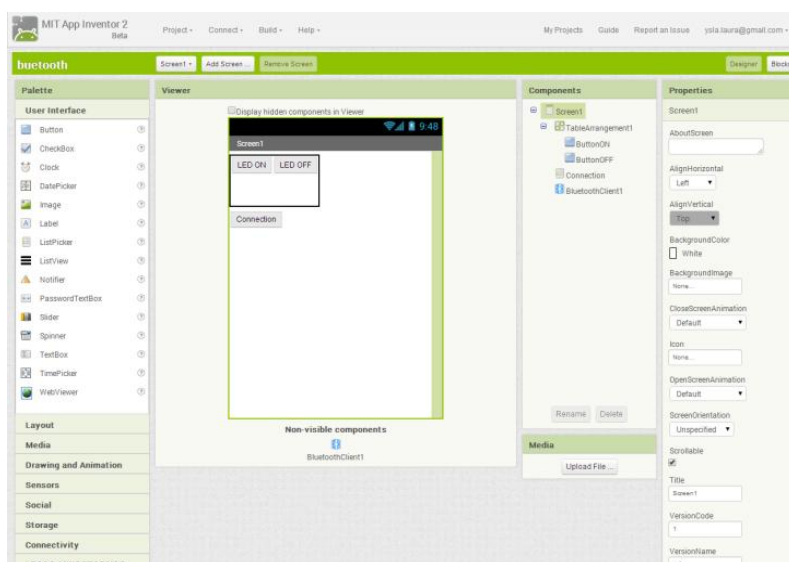


Figura 3.2 Interfície de disseny de l'*App Inventor*.

A la interfície que podem veure a la figura 3.2 a part de trobar tots els elements necessaris per a la nostra aplicació és on podem millorar tota l'estètica del nostre programa, és a dir, modificar la mida de les lletres, el tipus, el color, la ubicació, etc.



Una vegada seleccionats tots els elements necessaris per a la nostra aplicació hem de realitzar un programa que ens permeti interactuar amb els diferents elements, i per fer això hi ha una segona interfície que està basada en blocs de programació, tal i com podem veure a la figura 3.3.

Aquests blocs de programació són els encarregats de crear el programa, és a dir, cada bloc que creem s'encarregarà de la funció que li assignem. Això és possible gràcies a que en aquesta segona interfície trobem els elements seleccionats anteriorment a la part esquerra de la pantalla.

Quan pitgem cada element ens surten totes les funcions que poden dur a terme i hem d'anar seleccionant la més adient pel nostre programa. A part de les funcions predeterminades per cada element també disposem de set opcions més: *control*, *logic*, *math*, *text*, *list*, *colors*, *variables* i *procedures*.

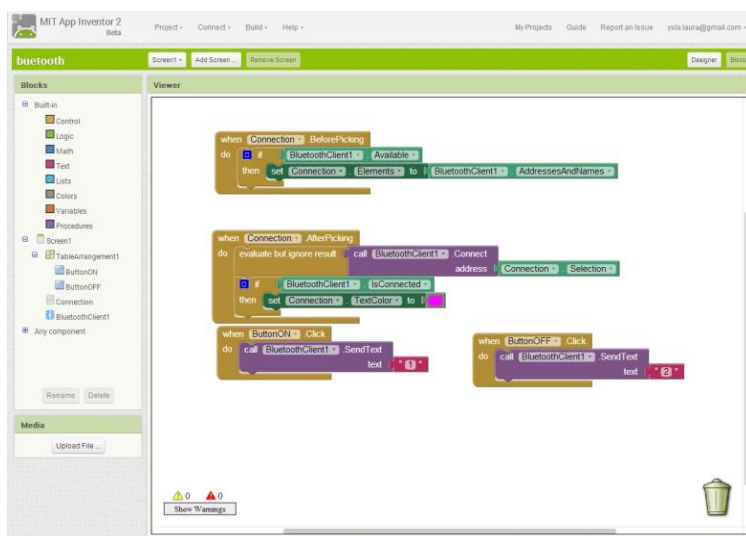


Figura 3.3 Interfície de blocs de l'App Inventor.

Un cop tenim la part de disseny i la dels blocs feta, podem connectar l'aplicació amb el mòbil per veure si funciona i si fa falta modificar alguna cosa. Un cop provada i modificada només en cas que alguna cosa fallés, podem construir l'aplicació i així poder-la tenir al nostre mòbil i utilitzar-la sempre vulguem.

Aquest programa ens permet crear aplicacions en funció de la dificultat que cadascú triï i també permet interactuar amb el nostre dispositiu físic, la placa *Arduino UNO*, mitjançant la utilització del *bluetooth*, per la qual cosa el nostre programa funcionarà en qualsevol mòbil que disposi de *bluetooth* i del sistema operatiu *Android*.

## L'HIVERNACLE INTEL·LIGENT



Gràcies a l'aplicació que crearem amb aquest entorn, l'*App Inventor* s'encarrega d'entrellaçar la pròpia aplicació amb els programes que controla *Arduino*, és a dir, ens permet crear una aplicació unint les dues plataformes utilitzades. Aquest entorn integrat de desenvolupament també ha fet possible crear una aplicació que qualsevol persona pugui entendre ja que l'aplicació és senzilla i pràctica i permet controlar tots els processos que vulguem que el nostre hivernacle dugui a terme.



## 4.-DISSENY I CONTRUCCIÓ DE L'HIVERNACLE

Un hivernacle és un espai tancat que s'utilitza a la producció de cultius a gran escala, i a cultivar una quantitat petita de plantes en aquest cas. Normalment està compost per una coberta exterior translúcida de vidre o plàstic, la qual permet el control de la temperatura, la humitat i d'altres factors ambientals per afavorir el desenvolupament de les plantes.

Aquesta estructura aprofita l'efecte produït per la radiació solar que, en travessar



la coberta, escalfa l'ambient a l'interior d'aquesta. Aquest procés produït pel sol, emet una radiació infraroja i aquesta no pot travessar la coberta quan retorna, i és per això que es queden atrapats a dins i produeixen escalfor.

Figura 4.1 Hivernacle d'un cultiu de mitja escala.

### 4.1- TIPUS DE PLANTES

Per plantar al nostre hivernacle, hem escollit plantes aromàtiques, ja que a casa meva sempre hem tingut debilitat per aquest tipus de plantes. Es tracta d'espècies que serveixen tant per cuinar, per fer infusions i sobretot per la bona olor que fan.

Les plantes o herbes aromàtiques són plantes molt utilitzades a la cuina mediterrània per les seves qualitats aromàtiques.



Figura 4.2 Planta de romaní.



Normalment s'utilitzen les fulles d'aquestes, tant si són fresques, com seques o deshidratades per tal de donar-li gust als menjars o per realçar les diverses olors dels plats culinàries, ja siguin crus o cuinats.

Per tal de cobrir totes les necessitats d'humitat (ambiental i del terra) i de temperatura de les diferents espècies vegetals, s'ha dissenyat un prototip amb dues parts diferenciades: una coberta i una altra descoberta. A la part coberta hem plantat menta i julivert, mentre que a la part exterior trobem romaní. D'aquesta manera tenim dos medis amb condicions de temperatura i humitat diferenciats i, en un futur, podem incorporar noves espècies aromàtiques a l'espai que considerem més adient.

### 4.2- DISSENY I MATERIALS

El disseny del meu hivernacle és una reproducció a escala d'un hivernacle estàndard. En quant a la base, és una caixa de plàstic, mentre que els que s'utilitzen al camp la seva base és la terra. Les dimensions de la maqueta d'aquest hivernacle són reduïdes ja que es tracta d'un prototip pensat per estar en un espai no gaire gran.



Figura 4.3 Base de l'hivernacle.

La finalitat d'aquest és poder tenir cura de les plantes més delicades, per la qual cosa les mides variaran en funció de quantes plantes sigui necessari cuidar. És per això que en aquest cas, al tractar-se d'un prototip, les mides són més reduïdes, ja



que aquestes són perfectes per tal de cuidar plantes aromàtiques.

Les seves reduïdes dimensions (53 x 34 x 14 cm) permeten una mobilitat major, ja que aquest no disposa d'un volum gaire gran i el seu pes no es gaire elevat. Com que la base és de plàstic es pot ajustar al gust de l'usuari permetent-li tenir una àmplia gama d'opcions a l'hora de construir-lo.

Com ja he comentat anteriorment, donat les diferents necessitats de reg, humitat i temperatura, hem vist la necessitat de dissenyar un prototip amb dos ambients diferenciats: un cobert i un altre descobert. La part coberta té una superfície de 1309 cm<sup>2</sup> i la part descoberta té una superfície de 448.8 cm<sup>2</sup>.

La part més gran està coberta per policristal<sup>6</sup> de 0.25 mil·límetres de gruix. El fet de cobrir una part permet controlar la temperatura, la humitat i altres factors ambientals per afavorir el desenvolupament de les plantes, ja que a la part descoberta les plantes no estan protegides. Aquesta part coberta té la funció de treballar com a medi selectiu de la transmissió per diverses freqüències espectrals, atrapar l'energia dins l'hivernacle i escalfar l'ambient interior. També serveix per evitar la pèrdua d'escalfor per convecció<sup>7</sup>.



La coberta d'aquesta part queda fixa per uns suports de PVC, material que proporciona una subjecció adequada de la coberta, és rígid i estable, condició que ha de complir per tal d'aguantar l'efecte de diferents fenòmens meteorològics, sempre i quan el nostre hivernacle es trobi situat a l'exterior de casa i no a l'interior. El diàmetre de les columnes és de 3.2 centímetres i tenen una longitud de 40 centímetres.

Figura 4.4 Columnes de PVC.

<sup>6</sup> Pertanyent a la família dels polímers. Són macromolècules compostes per un o varis elements químics que es repeteixen en una cadena.

<sup>7</sup> Propagació de la calor en fluïts i líquids pel moviment de les seves partícules produïdes per les diferents densitats.

## L'HIVERNACLE INTEL·LIGENT



La part descoberta té dos apartats, un dedicat al romaní i l'altra dedicat al controlador. A la separació del controlador trobem una secció de control, ja que trobem el *LED* que mostra que el controlador funciona i el polsador.



## 5.-DISSENY DEL SISTEMA DE CONTROL AUTOMÀTIC

### 5.1-SISTEMES DE CONTROL

Un sistema de control automàtic és el conjunt d'elements tècnics que té com a objectiu aconseguir que una màquina o un procés realitzi les seves funcions amb una intervenció humana mínima. Només s'ha d'intervenir per tal d'introduir les ordres de funcionament en el sistema de control per tal que aquest executi de manera adequada les fases de treball.

La tecnologia de control engloba tots els procediments i sistemes que permeten l'automatització de màquines, aparells i processos de fabricació. L'automatització es troba a diferents sectors econòmics: indústria, agricultura, ramaderia, pesca, serveis, comunicacions, domòtica, transports, etc.



Figura 5.1 Imatge d'un sistema de reg automàtic.

A l'agricultura serveix per controlar la temperatura, la humitat, sistemes automàtics de reg, que són justament els automatismes que trobem en aquest treball.

El nostre hivernacle, per tant, pot ser un sistema automàtic capaç de controlar un conjunt de variables ambientals (temperatura i humitat) que garanteixen l'òptim desenvolupament de les nostres plantes aromàtiques.





## 5.1.1- TIPUS DE SISTEMA DE CONTROL

En funció de la forma de treballar de l'automatisme trobem dos tipus de control: control en llaç obert i control en llaç tancat.

El **control en llaç obert** es caracteritza perquè, un cop activat, realitza el procés durant un temps prefixat, amb independència del resultat obtingut. El sistema de control no supervisa el resultat de la seva acció a la sortida del sistema. Són els sistemes més senzill i econòmics, i normalment realitzen accions molt simples.

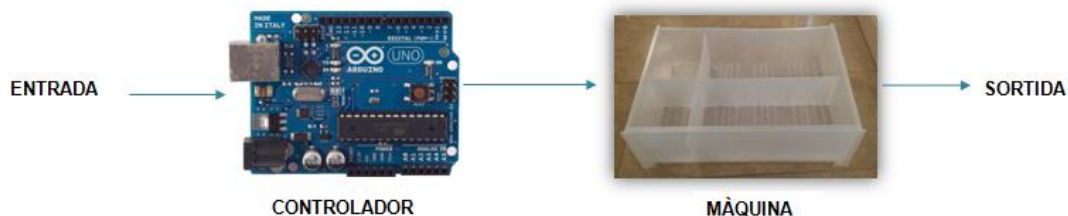


Figura 5.2 Diagrama de blocs. Funcionament d'un sistema de control automàtic en llaç obert.

Els sistemes de **control en llaç tancat** estableixen una supervisió sobre la variable de sortida, actuant fins a obtenir el resultat desitjat. En aquests sistemes, un cop donada l'ordre per iniciar el procés, el resultat de la sortida és analitzat i, si no compleix una determinada consigna, el dispositiu de control manté el procés actiu fins assolir el que estableix la consigna.

Un sistema de llaç tancat es caracteritza per l'acció de sortida sobre la d'entrada, ja que mitjançant un procés continu de realimentació, que consisteix a comparar l'estat de la sortida amb el de la consigna<sup>8</sup>, es modifica la resposta de l'automatisme d'acord amb el resultat de la comparació.

<sup>8</sup> La consigna és el valor de referència que es desitja obtenir com a resultat del sistema automàtic.

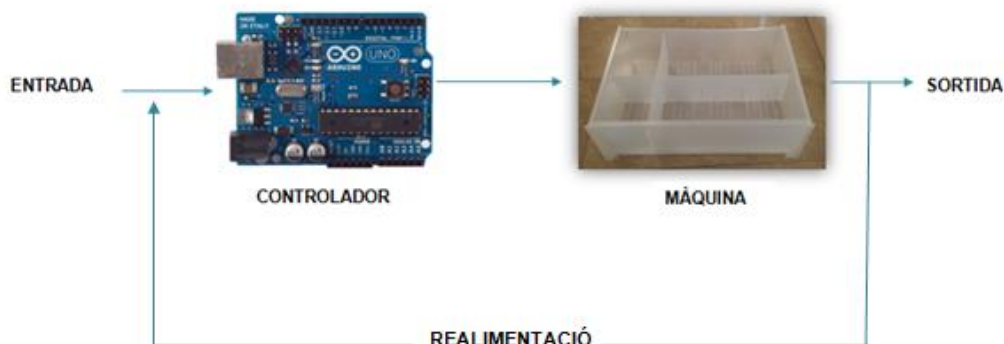


Figura 5.3 Diagrama de blocs. Funcionament d'un sistema de control automàtic en llaç tancat.

## 5.2- ESPECIFICACIONS DE DISSENY DE L'HIVERNACLE INTEL·LIGENT

En el disseny del nostre hivernacle intel·ligent, s'automatitzaran els següents processos:

### **1. Adquisició i monitorització de la temperatura exterior.**

Gràcies a un sensor de temperatura situat a la part exterior, podem visualitzar la temperatura d'aquesta zona tant al monitor sèrie<sup>9</sup> com a la pantalla del mòbil mitjançant la nostra aplicació.

### **2. Adquisició i monitorització de la temperatura i humitat interior.**

A partir d'un sensor de temperatura i humitat situat a la part interior del nostre hivernacle intel·ligent, rebem els valors de la temperatura i la humitat i podem visualitzar aquestes dades ja sigui al nostre ordinador (pantalla del monitor sèrie) o a la nostra aplicació.

<sup>9</sup> Eina d'Arduino que permet visualitzar a la pantalla del nostre ordinador les dades obtingudes gràcies a aquest sensor.



### 3. Sistema de reg automàtic.

La bomba d'aigua està programada a partir de les dades que obtindrem amb el nostre sensor de humitat casolà. Aquest sensor farà que aquest automatisme s'activi quan la humitat que registra és inferior al valor de la consigna preestablert.

### 4. Senyalització lluminosa de processos.

Cada sensor es relaciona amb els *LEDs*, és a dir, tots s'encenen per informar d'alguna cosa. El *LED* verd exterior s'encén per informar-nos de que la placa *Arduino UNO* funciona correctament. El *LED* multicolor ens informa de la temperatura exterior a partir d'una escala de colors<sup>10</sup>. El *LED* blanc exterior s'activa quan es fa fosc. El *LED* verd interior s'activa quan la temperatura interior és inferior a 20 °C, i el vermell s'activa quan aquesta és superior a 20°C. Per últim, el *LED* groc s'activa i parpelleja quan la humitat és inferior al 50%.

### 5. Il·luminació temporitzada amb accionament manual.

S'encén el *LED* blanc interior quan accionem el polsador. Aquest es manté encès durant 15 segons.



Figura 5.4 Imatge del nostre hivernacle intel·ligent.

## 5.3- ELEMENTS DEL NOSTRE SISTEMA DE CONTROL

A continuació descriuré, de forma ordenada, els elements que he utilitzat per dissenyar i implementar l'hivernacle intel·ligent.

<sup>10</sup> Quan més vermellós és, més calor i quan més verdós, més fred.



## 5.3.1-ELEMENTS DE COMANDAMENTS MANUALS

L'element de comandament manual que s'ha utilitzat en el nostre prototip és un polsador (figura 5.4). En concret, la funció d'aquest polsador és accionar un *LED* blanc situat a la part coberta. En definitiva, aquest element ens permet encendre i apagar el llum interior de forma manual i de forma temporitzada.



Figura 5.5 Polsador emprat a l'hivernacle.

## 5.3.2-SENSORS

Els sensors són elements de control que permeten la connexió, desconnexió i comandament general de les instal·lacions sense la intervenció directa de l'operari.

En el nostre treball trobem quatre sensors: el sensor de temperatura LM35, el sensor de llum, el sensor de temperatura i humitat DHT11 i el sensor de humitat casolà. Aquests quatre ens permeten en tot moment saber quina temperatura i humitat .



Figura 5.6 Sensor de temperatura i humitat.

El sensor DHT11 és un sensor bàsic de humitat i temperatura de baix cost. Utilitza un sensor de capacitat per mesurar la humitat i un termistor per mesurar la temperatura de l'aire que l'envolta. Està dissenyat per mesurar temperatures entre 0 i 50 °C amb una precisió de 2 °C i per mesurar humitat entre 20% i 80% amb una precisió d'un 5% amb períodes de mostreig d'1 segon.

El sensor de temperatura LM35 és un sensor bàsic de temperatura. Utilitza un termistor per tal de mesurar la temperatura. Està dissenyat per mesurar temperatures entre 0 °C i 100 °C. Té una precisió de 0.5 °C.



Figura 5.7 Sensor de temperatura.



L'altre sensor que utilitzem és el sensor de humitat casolà. Està constituït per dues barres d'acer de 15 centímetres de llargària, i recoberts 10 centímetres per un material aïllant. Aquestes dues barres es troben inserides dins d'un tros de porexpà. La part que sobresurt d'aquesta mena de suport està soldada a uns cables, per tal que aquests es comuniquin amb la placa, ja que aquest sensor ens ofereix la humitat de la terra de la part coberta.

Figura 5.8 Sensor de humitat casolà.

El sensor de llum és un dispositiu electrònic semiconductor. Té la propietat de variar la seva resistència en funció de la llum que rep. Quanta més llum rep, menor és la seva resistència, i quan menys llum, major resistència. La seva resistència varia entre valors de  $100 \Omega$  a  $10M \Omega$ .

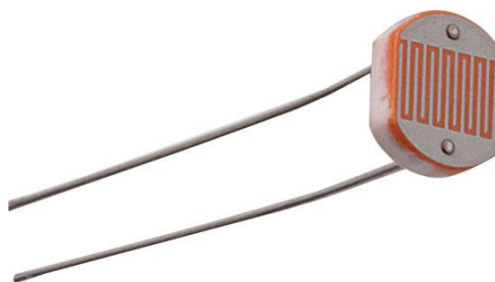


Figura 5.9 Sensor de llum.

### 5.3.3- CONTROLADORS

El controlador és el dispositiu característic del sistema que determina i executa el control del procés per al qual està programat. És, per tant, el cervell del sistema de control.

Els sistemes de control en llaç tancat, constitueix el sistema de tancament de la informació del procés, i estableix la manera com s'han de combinar les entrades d'informació per tal d'activar les sortides del sistema.



Figura 5.10 Placa *Arduino* UNO.

En aquest cas el nostre controlador, com ja hem pogut veure anteriorment al capítol 1, és la placa *Arduino* UNO, ja que aquesta és la que executa tots els processos que du a terme l'hivernacle. Es tracta d'un controlador electrònic.

## 5.3.4- ACTUADORS

Un actuator és un dispositiu que permet transformar una magnitud elèctrica en una altra forma d'energia o que permet l'amplificació d'un mateix tipus d'energia.

L'actuator utilitzats per a la maqueta és una bomba d'aigua. Aquest es pot programar per tal de que funcionin durant un període de temps, quan un altre element dona l'ordre o es poden accionar manualment mitjançant el nostre *smartphone*.



Figura 5.11 Actuator emprat: bomba d'aigua.

## 5.3.5-ELEMENTS DE COMUNICACIÓ: BLUETOOTH



Figura 5.12 Mòdul *bluetooth*.

El mòdul de *bluetooth* HC-06 ofereix una millor relació de preu i característiques degut a que a més de rebre connexions des d'un PC o tableta, també és capaç de generar connexions cap a altres dispositius *bluetooth*. Això ens permet connectar dos mòduls *bluetooth* i formar una connexió punt a punt per transmetre dades entre un microcontrolador i un dispositiu.



Finalment, per acabar aquest capítol, trobem una taula resum on expliquem quines entrades hi ha, les sortides que controlen, la funció que du a terme cada element i com es controla mitjançant la nostra aplicació creada amb l'*App Inventor*.

ENTRADA	SORTIDA	FUNCIÓ	APP INVENTOR
	LED verd	Assegurar que el controlador funciona.	
Polsador	LED blanc interior	Permet encendre el LED en qualsevol moment.	Permet encendre el mateix LED a partir de l'aplicació.
LDR	LED blanc exterior	S'encén quan es fa de nit.	Es canvia el fons de l'aplicació en funció de la llum.
Sensor de temperatura	LED multicolor	Canvia la seva tonalitat en funció de la temperatura que fa. Si la temperatura és elevada aquest té un color vermellós i si és baixa té un color blavós.	Permet visualitzar la temperatura en tot moment.
Sensor de temperatura i humitat	LED groc LED verd LED vermell	S'activa el LED groc quan el valor de la humitat és inferior al que hem establert. El altres dos s'activen o bé quan la temperatura és superior a la establerta, o inferior.	Podem visualitzar els valors que ens proporciona aquest sensor al mòbil.
Sensor d'humitat	Bomba d'aigua	Quan la humitat és inferior al valor establert s'activa la bomba d'aigua.	Permet activar la bomba manualment a partir de l'aplicació.

Figura 5.13 Taula resum dels automatismes emprats.



## 6.-PROGRAMACIÓ DEL SISTEMA DE CONTROL AUTOMÀTIC

En aquest capítol trobem el funcionament del nostre programa realitzat mitjançant *Arduino*, ja que com hem pogut anar veient durant el treball, és el programa que hem utilitzat ja que és el més complet i més adient per dur a terme tots els processos o funcions que volem que el nostre hivernacle sigui capaç de fer. El codi de programa es pot consultar íntegrament a l'annex 2.

El nostre programa és divideix en 8 parts, ja que el programa final es tracta de la suma de diferents programes individuals els quals s'encarreguen d'un procés determinat.

Aquests 8 subprogrames corresponen als elements i automatismes que hem anat explicant als capítols anteriors.

El primer pas que hem de dur a terme perquè aquest programa funcioni és definir les llibreries que utilitzarem i definir totes les variables (*LEDs*, sensor, polsador, mòdul *bluetooth* CH-06). Tot seguit hem de definir dins de l'estructura *void set up()* quins elements funcionen com a sortida i quins com a entrades. Seguidament dins de la funció *void loop()* definim el nom de cada subprograma per tal de que el programa resultant llegeixi cada programa periòdicament amb una diferència de 5 milisegons.

El primer subprograma s'anomena **sensor**, i aquest s'encarrega d'enviar les dades obtingudes pels sensor LM35, DHT11 i pel sensor de humitat casolà a la nostra aplicació i així permetre'ns poder visualitzar la temperatura en la nostra aplicació. Proporciona la temperatura interior, la humitat interior, temperatura exterior i humitat de la terra.

El segon subprograma s'anomena **bluetooth\_app**. Aquest programa conté el conjunt d'instruccions necessàries per tal de configurar i comunicar el mòdul *bluetooth* amb la nostra aplicació *Android*. D'aquesta manera podem enviar i rebre





dades entre la placa i el nostre *smartphone*, i per tant, podem activar i desactivar elements de la nostra maqueta a distància.

El tercer subprograma s'anomena **led\_verd**, i és el subprograma més simple, ja que la funció d'aquest només és activar el *LED* verd per tal d'informar-nos de que la placa funciona correctament. Aquest *LED* està connectat al pin digital 13.

El quart subprograma s'anomena **polsador**. Aquest subprograma s'encarrega de relacionar el *LED* blanc interior amb el polsador, és a dir, ens permet activar aquest *LED* durant 15 segons quan pressionem el polsador. En aquest cas el polsador el connectem al pin digital 7 i el *LED* blanc al pin digital 12.

El cinquè subprograma s'anomena **sensor\_llum** i s'encarrega d'activar el *LED* blanc exterior quan el valor obtingut de la llum és inferior al valor preestablert al nostre programa, és a dir, està programat perquè quan es faci de nit aquest sensor baixi la seva resistència fent així que el *LED* s'activi i ens il·lumini l'hivernacle. Hem connectat aquest sensor al pin analògic A0 i el *LED* blanc al pin digital 11. Aquest programa també és l'encarregat de canviar la pantalla de la nostra aplicació, és a dir, és el responsable de que la nostra aplicació, a partir d'una imatge, ens permeti saber si es de dia o de nit.

El sisè subprograma s'anomena **sensor\_LM35**. Aquest s'encarrega de proporcionar-nos la temperatura de la part exterior de l'hivernacle. A part de permetre'ns visualitzar la temperatura d'aquesta part també activa un *LED* multicolor, el qual té la funció d'informar-nos de quina temperatura fa a partir d'una escala de colors, ja que està programat perquè si la temperatura és de gairebé 40 °C el color sigui vermellós i si la temperatura és propera als 5 °C el color sigui verdós. En aquest subprograma connectem el sensor LM35 al pin analògic A1 i el *LED* multicolor als pins digitals 5 i 6.

El setè subprograma s'anomena **sensor\_temperatura\_humitat** i s'encarrega d'informar-nos de la temperatura i la humitat que hi ha a la part coberta. A part de permetre'ns veure aquestes dades al monitor sèrie i a la nostra aplicació, també funciona amb tres *LEDs* per tal d'informar-nos aproximadament de la temperatura que fa en aquesta part. El subprograma el que fa és establir que quan la



temperatura que el sensor DHT11 recull sigui inferior a 20 °C s'activi un *LED* verd. Si aquesta temperatura és superior a aquests 20 °C, el *LED* verd deixa de funcionar i s'activa un *LED* vermell. També ens informa de la humitat, és a dir, ens informa quan el valor de la humitat és inferior al 50%, ja que si això passa s'activa el *LED* groc i comença a parpellejar. Aquest sensor l'hem de connectar al pin digital 4. El *LED* groc està connectat al pin digital 8, el *LED* vermell al pin digital 8 i el *LED* verd al pin digital 10.

Finalment acabem el nostre programa amb el subprograma **sensor\_humitat**. Aquest s'encarrega d'activar la bomba d'aigua quan el valor que rep és inferior o superior al valor preestablert anteriorment. És a dir, si la humitat és superior al valor que hem establert, la bomba es manté apagada, però si aquest valor és inferior, la bomba s'activa. També s'encarrega de permetre'ns saber la humitat de la terra de la part coberta. En aquest últim subprograma hem connectat el sensor casolà al pin analògic A2 i la bomba d'aigua a pin digital 1.

AUTOMATISME	POLSADOR	SENSOR DE TEMPERATURA	SENSOR D'HUMITAT	SENSOR DE TEMPERATURA I HUMITAT	SENSOR DE LLUM
NUMERO PIN ANALOGIC		1	2		0
NUMERO PIN DIGITAL	7			4	

LED ACTIVAT	LED blanc interior	LED multicolor exterior	Bomba d'aigua	LED verd, LED groc, LED vermell interiors	LED blanc exterior
NUMERO PIN DIGITAL	12	5 i 6	1	8, 9 i 10	11

Figura 6.1 Taula resum de les connexions dels automatismes emprats i de les connexions dels elements controlats per aquests automatismes.

## L'HIVERNACLE INTEL·LIGENT



A la figura 6.1 podem veure una taula resum de on es connecta cada sensor i on es connecten els elements controlats per aquests sensors.

Per finalitzar el nostre programa, cal repassar-lo i verificar que no hi ha cap problema d'escriptura ni cap error a l'hora de relacionar l'aplicació amb aquests. Un cop corregits els possibles errors cal realitzar el muntatge del nostre circuit a la maqueta, i un cop tots els elements estan col·locats i connectats al pin corresponent, només hem de carregar el nostre programa al nostre microcontrolador i assegurar-nos de que cada element funciona correctament i fa la funció que li toca.



## 7.-APLICACIÓ: SMART GREENHOUSE

En aquest capítol està explicat el funcionament de la nostra aplicació, anomenada "Smart greenhouse", ja que el nom és una referència clara a la funció que permet realitzar, és a dir, permetre que el nostre hivernacle sigui intel·ligent.

L'objectiu d'aquesta aplicació és permetre'ns controlar la temperatura i la humitat de les nostres plantes en tot moment a part de poder activar i desactivar certs automatismes, i tot això d'una forma senzilla i còmoda.

### 7.1- DISSENY DE LA PANTALLA

Per que la nostra aplicació funcioni hem d'utilitzar els següents elements de disseny de pantalla: una *table arrangement*<sup>11</sup>, quatre botons, un *label*<sup>12</sup>, una *list picker*<sup>13</sup>, la connexió *bluetooth* i un rellotge.

A la figura 7.1 podem observar el disseny de pantalla de la nostra aplicació. Com es pot veure, hem encerclat en verd el conjunt de botons que ens permetran controlar l'encesa i tancament dels llums situats a l'interior de l'hivernacle i l'accionament de la bomba d'aigua. Aquests botons estan continguts dins l'element anomenat *table arrangement*, que permet situar-los ordenats en aquest punt de la pantalla.

Sota el conjunt de botons de control, trobem un àrea dins la qual visualitzarem les dades de temperatura (exterior i interior) i humitat interior. Per realitzar la visualització hem utilitzat un component anomenat *label*, dins del qual crearem la estació meteorològica del nostre hivernacle, ja que ens permet visualitzar les dades obtingudes a partir dels sensors després d'haver-nos connectat.

<sup>11</sup> Es tracta d'un organitzador de la pantalla, és a dir, de la part que l'usuari veu i manipula.

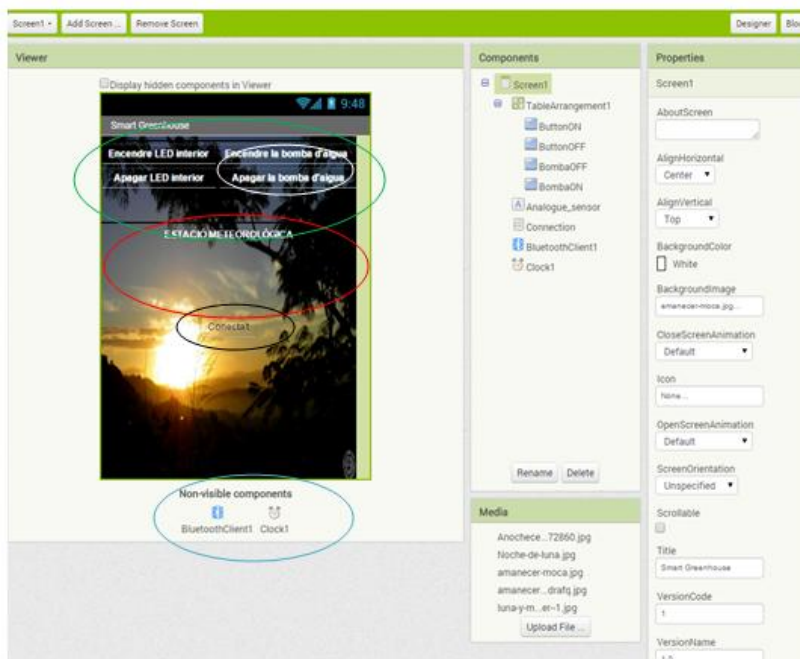
<sup>12</sup> Permet la visualització de les dades obtingudes mitjançant el sensor utilitzats.

<sup>13</sup> Es tracta d'un botó que permet la connexió del pin *bluetooth* amb el *bluetooth* del mòbil per tal de que l'aplicació funcioni i es transmetin les dades.



A la part inferior de la pantalla trobem el botó de connexió encerclat en negre. Aquest és el responsable de que puguem connectar el nostre *smartphone* amb el nostre mòdul *bluetooth*.

Altres elements necessaris per tal que la nostra aplicació funcioni són el rellotge i la connexió *bluetooth*. Aquests dos components són els responsables de que hi pugui haver comunicació entre la nostra aplicació i el nostre microcontrolador



*Arduino*.

Figura 7.1 Imatge del disseny de la pantalla de dia de la nostra aplicació.

Aquesta aplicació, controla el *LED* blanc interior i la bomba d'aigua després d'haver-nos connectat, és a dir, després de pulsar el botó de "connecta't", posar la contrasenya i seleccionar el mòdul *bluetooth* CH-06.

## 7.2-PROGRAMACIÓ DE L'APLICACIÓ

Un cop tenim tots els elements seleccionats i al seu lloc, hem de dissenyar el programa a l'apartat de diagrama de blocs. En aquest apartat hem establert que, quan pitgem el botó de "connecta't", les lletres canviaran de color per tal d'informar-nos que la nostra connexió ha estat realitzada sense cap mena de problema.



També hem establert que quan el *bluetooth* està connectat, l'element *label* rep les dades que al nostre programa hem establert que s'havien d'enviar, i les podem visualitzar a la nostra estació meteorològica.

També hem programat els quatre botons. Hem establert que si pitgem el botó d'encendre el *LED*, s'envia un 1 al mòdul *bluetooth*, la qual cosa permetrà que el *LED* s'encengui. Si s'envia un 2, el *LED* s'apagarà. Amb la bomba d'aigua passa exactament el mateix, però en aquest cas canviem de números; si enviem un 3 la bomba s'encén i si enviem un 4 la bomba s'apaga.

A continuació, a la figura 7.2 podem observar els blocs que hem utilitzat per fer funcionar la nostra aplicació.

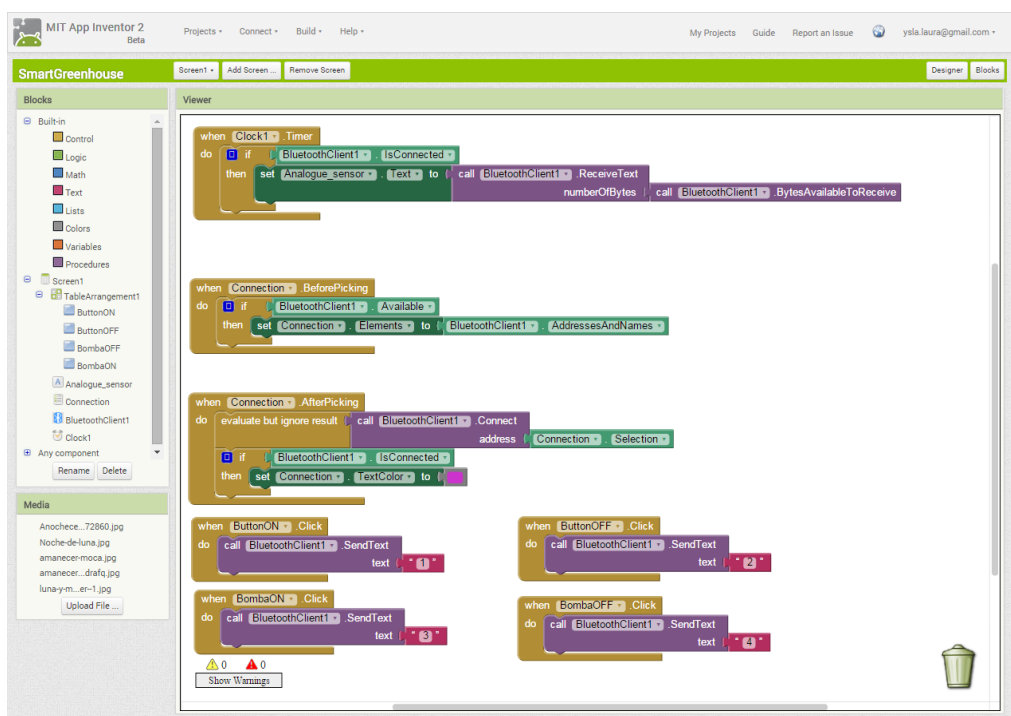


Figura 7.2 Imatge dels blocs de la nostra aplicació.

Finalment, un cop ens hem assegurat de que tenim tot ben fet, és a dir, que tots els blocs funcionen adequadament i que tenim seleccionats tots els elements necessaris cal fer una comprovació prèvia a crear l'aplicació per tal de veure si tot funciona correctament i que no hi ha cap problema entre els dos programes que hem utilitzat (*Arduino* i *App Inventor*), i un cop assegurat que tot funciona adequadament podem crear l'aplicació.



## CONCLUSIONS

Després d'haver realitzat aquest treball podem concloure que:

1.- La plataforma *Arduino* UNO ha estat la més adient per realitzar aquest treball degut al seu baix cost econòmic, pel seu llenguatge de programació, la seva versatilitat, l'elevat nombre d'entrades i sortides i, sobre tot, per la possibilitat de connectar el nostre prototip a una aplicació *Android* mitjançant *bluetooth*.

2.-El llenguatge de programació *Java Script* ens ha permès crear un programa amb molts elements i funcions de forma relativament senzilla. També ens ha facilitat la lectura d'aquest, ja que permet programar de forma clara i precisa a la vegada que pots modificar qualsevol comandament sempre que l'usuari vulgui.

3.-*App Inventor* ha estat un entorn integrat de desenvolupament òptim per tal de realitzar la nostra aplicació, ja que ens ha permès crear una aplicació i relacionar-la amb el nostre microcontrolador de forma senzilla a la vegada que ens ha ofert una àmplia gama d'eines per fer la nostra aplicació fàcil d'utilitzar.

4.-El fet d'escollir una base de plàstic com a eina de treball, juntament amb els suports de PVC i la coberta de policristal ens ha permès una fàcil instal·lació dels nostres automatismes, a la vegada que ens ha permès un fàcil desplaçament d'aquest hivernacle.

5.-El nostre hivernacle intel·ligent ens ha permès fer un ampli estudi de les temperatures i les humitats que afecten les nostres plantes. Ens ha permès també poder controlar les funcions de dues maneres diferents, donant-nos més joc a l'hora de crear el programa.

6.-Hem pogut automatitzar 5 processos i relacionar-los de forma ràpida i senzilla amb la nostra aplicació gràcies a l'ús del mòdul *bluetooth*. Cal tenir en compte que



aquests processos s'han pogut automatitzar gràcies a la suma de tots els elements, no només pel mòdul *bluetooth*.

7.-El fet d'organitzar el nostre programa en 8 subprogrames ens ha permès anar provant un a un aquest subprogrames, fent així que la instal·lació i la recopilació del programa fossin més fàcils de dur a terme.

8.-Seguir una estructura lògica i ser ordenats a l'hora de realitzar el programa ha estat l'element clau per aconseguir que aquest programa funcionés i realitzés totes les funcions que volíem automatitzar.

9.-El disseny de la nostra aplicació no ha estat gaire difícil de realitzar, ja que com hem pogut veure al llarg del treball, no hem necessitat gaires components per tal de fer que aquesta funcionés adequadament. *App Inventor* ens ha permès crear una aplicació fàcil d'utilitzar i a la vegada ens ha permès fer-la entenedora a l'hora d'interactuar amb ella.





### PROPOSTES DE MILLORA

La primera proposta de millora que podem fer respecte a aquest projecte és la utilització del *bluetooth*. Com ja sabem, per aconseguir que els dos programes utilitzats interactuessin entre ells hem utilitzat un pin *bluetooth*, i això ha reduït el radi d'acció de la nostra aplicació, ja que aquest funciona només dins de casa, és a dir, no ens permet saber l'estat de les plantes mentre estem fora de casa, com podria ser a l'oficina.

Per tal d'augmentar el radi d'acció de la nostra aplicació només caldria canviar d'eina relacionant, és a dir, en comptes d'utilitzar el *bluetooth*, utilitzar algun altre dispositiu que funcionés amb *wifi*, la qual cosa augmentaria el radi d'acció fent l'aplicació més efectiva, tot i que l'aplicació funciona perfectament dins de casa.

L'altra proposta de millora és aconseguir que la nostra aplicació, juntament amb les dades que proporciona la placa *Arduino UNO*, es comuniqui via *Twitter*, és a dir, que a part de poder visualitzar les dades obtingudes a la nostra aplicació, que també puguem saber quan necessiten aigua, o més llum o menys a partir d'una publicació a *Twitter*.



## LLISTA DE REFERÈNCIES

### BIBLIOGRAFIA

- TOJEIRO CALAZA, Germán. *Taller de Arduino. Un enfoque práctico para principiantes*. Barcelona; Marcombo, 2014.
- LUIS DDURÁN, José; GÁMIZ, Juan; BARGALLÓ, Ramón; DOMINGO, Joan; MARTÍNEZ, Herminio; MORÓN, Juan. *Electrotècnia. Instal·lacions elèctriques i automàtiques*. Barcelona; Marcombo, 2009.
- OXER HUGH BLEMINGS, Jonathan. *Practical Arduino. Cool Projects for Open Source Hardware*. Nova York; ATTiCA, 2009.
- EVANS, Brian. *Beginning Arduino Programming. Writing Code for the Most Popular Microcontroller Board in the World*. Nova York; ATTiCA, 2009.
- WILCHER, Don. *Learn Electronics with Arduino*. Nova York; ATTiCA, 2009.
- RILEY, Mike. *Programming Your Home. Automate with Arduino, Android, and Your Computer*. Dallas; The Pragmatic Programmers, 2012.

### WEBGRAFIA

- *Arduino* [en línia]. <<http://arduino.cc/>> [consulta: 07.06.2014].
- *Wiring, Wikipedia* [en línia]. <<http://ca.wikipedia.org/wiki/Wiring>> [consulta: 07.06.2014].
- *Processing 2* [en línia]. <<http://www.processing.org/>> [consulta: 07.06.2014].
- *Overview, Arduino* [en línia]. <<http://arduino.cc/en/Main/ArduinoBoardUno>> [consulta: 07.06.2014].
- *ASCII, Wikipedia* [en línia]. <<http://es.wikipedia.org/wiki/ASCII>> [consulta: 15.06.2014].



- *Reference, Arduino* [en línia]. <<http://arduino.cc/en/Reference/HomePage>> [consulta: 15.06.2014].
- Definició microcontrolador, *Microcontroladores géricos* [en línia] <<http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>> [consulta: 19.06.2014].
- ICSP, *Wikipedia* [en línia] < [http://en.wikipedia.org/wiki/In-circuit\\_serial\\_programming](http://en.wikipedia.org/wiki/In-circuit_serial_programming)> [consulta: 23.06.2014].
- *App Inventor, Wikipedia* [en línia] <[http://ca.wikipedia.org/wiki/App\\_Inventor](http://ca.wikipedia.org/wiki/App_Inventor)> [consulta: 29.06.2014].
- Hivernacle, *Innatia* [en línia] < <http://www.innatia.com/s/c-huerta-organica/a-invernaderos-caseros.html>> [consulta: 05.08.2014].
- Hivernacle, *Wikipedia* [en línia] <<http://es.wikipedia.org/wiki/Invernadero>> [consulta: 07.10.2014].
- Convecció, *Wordreference* [en línia] <<http://www.wordreference.com/definicion/convecci%C3%B3n>> [consulta: 07.10.2014].
- Entrades, *Entradas y sartides* [en línia] <[http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMh1/PAGINA%20PRINCIPAL/PLC/ESTRUCTURAS/ESTRUCTURA%20INTERNA/SECCION%20DE%20ES/seccion\\_de\\_es.htm#Entradas digitales](http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMh1/PAGINA%20PRINCIPAL/PLC/ESTRUCTURAS/ESTRUCTURA%20INTERNA/SECCION%20DE%20ES/seccion_de_es.htm#Entradas%20digitales)> [consulta: 07.10.2014].
- *MultipleScreen, Youtube* [en línia] <<https://www.youtube.com/watch?v=Myvyo0TNAYw>> [consulta:08.10.2014].
- *LDR, Arduteka* [en línia] <<http://www.arduteka.com/2011/11/componentes-el-sensor-ldr/>> [consulta:08.10.2014].
- Sensor de temperatura, *medirtemperatura.com* [en línia] <<http://medirtemperatura.com/sensor-temperatura.php>> [consulta:08.10.2014].
- Sensor de temperatura i humitat, *Proyectos Arduino* [en línia] <<http://www.proyectosarduino.com.ar/sensor-de-temperatura-y-humedad-dht11/>> [consulta:0.10.2014].



- *LED*, *Master Magazine* [en línia] <<http://www.mastermagazine.info/termino/5554.php>> [consulta:08.10.2014].
- *Token*, *Twitter* [en línia] < <http://arduino-tweet.appspot.com/>> [consulta: 18.10.2014].
- *Plantes aromàtiques*, *Wikipedia* [en línia] <[http://es.wikipedia.org/wiki/Hierba\\_arom%C3%A1tica](http://es.wikipedia.org/wiki/Hierba_arom%C3%A1tica)> [consulta:13.12.2014].
- *Propietats dels materials*, *Tecnologia Industrials* [en línia] <<http://www.xtec.cat/~jsolson7/batx/materials1/propiedades.htm>> [consulta:13.12.2014].
- *Scratch for Arduino* [en línia] <<http://s4a.cat/>-17.12.2014> [consulta: 17.12.2014].
- *Plantes aromàtiques*, *Hogarutil* [en línia] <<http://www.hogarutil.com/jardinaria/mantenimiento/huerta/201212/plantas-aromaticas-interior-exterior-17905.html>> [consulta:17.12.2014].
- *Sensor de temperatura i humitat*, *Internet de las cosas* [en línia] <<http://www.internetdelascosas.cl/2014/07/08/midiendo-temperatura-y-humedad-con-arduino-y-el-sensor-dht11/>> [consulta: 20.12.2014].
- *Bluetooth i App Inventor*, *gerdsLAB* [en línia] <[http://gerdslab.com/es/bluetooth\\_appinventor1](http://gerdslab.com/es/bluetooth_appinventor1)> [consulta: 23.12.2014].
- *DHT11*, *DHT11 Humidity & Temperature Sensor* [en línia] <<http://www.micro4you.com/files/sensor/DHT11.pdf>> [consulta: 02.01.2015].
- *Sensor de temperatura*, *LM35 Precision Centigrade Temperature Sensors* [en línia] < <http://www.ti.com/lit/ds/symlink/lm35.pdf>> [consulta: 02.01.2015].
- *Sensor de llum*, *Light dependent resistors* [en línia] <<http://www.ti.com/lit/ds/symlink/lm35.pdf>> [consulta: 02.01.2015].
- *BD138NPN*, *PNP Silicon Transistor* [en línia] <<http://www.farnell.com/datasheets/309451.pdf>> [consulta: 04.01.2015].

## L'HIVERNACLE INTEL·LIGENT



- Components *App Inventor*, *App Inventor* en Español [en línea]  
<<https://sites.google.com/site/appinventormegusta/guias-visuales/componentes/componentes-basicos/listpicker>>  
[consulta: 05.01.2015].

**ANNEXOS**

## ANNEX 1: PRESSUPOST DEL TEBALL



## 1.-COSTOS D'ENGINYERIA

### 1.1-CÀLCUL DEL PREU HORA

Amb l'objectiu de realitzar el càlcul aproximat del preu/hora, suposarem que el projecte ha estat realitzat per un únic enginyer. Per realitzar aquest càlcul s'han de tenir en compte els costos directes i els indirectes.

#### 1.1.1-COSTOS DIRECTES

Els costos directes es defineixen com: "la suma dels costos de material, mà d'obra i equip necessari per a la realització d'un procés productiu".

Nombre d'enginyers	1
Salari anual enginyer	46.000 €
Nº d'hores anuals treballades	1.700 h
Hores venudes	60%
Nº d'hores venudes	1020 h/any
Preu/hora	46 €

Figura 1.1 Variables a tenir en compte per al càlcul dels costos directes.

Per fer aquest treball s'ha tingut en consideració que està fet només per un enginyer el qual cobra un salari anual de 46.000 €. A partir de les dades que podem observar a la taula s'ha calculat que el preu/hora d'un enginyer és de 46€.





## 1.1.2-COSTOS INDIRECTES

Els costos indirectes són aquells que deriven de l'execució d'un procés constructiu del qual deriva un producte. No es té en compte la mà d'obra, els materials i la maquinaria.

Cost anual lloguer del local	3.600 €
Equips d'instrumentació	2.400 €
Equips informàtics	1.500 €
Material oficina	1.400 €
Despreses oficina	1.200 €
<b>Total costos indirectes</b>	<b>10.100 €</b>

Figura 1.2 Variables a tenir en compte per al càlcul dels costos indirectes.

De la segona taula podem calcular el preu hora dels costos indirectes. Aquest preu és de 6 €/h.

## 1.1.3-PREU TOTAL

El preu total és la suma del cost directe més el cost indirecte. El preu total del projecte puja a 52 €.

## 1.2-COSTOS DERIVATS DEL DISSENY

A continuació podem observar una taula en la qual trobem el preu de tots els elements emprats al Treball de Recerca.



Placa <i>Arduino</i> UNO	20.20 €
Pin bluetooth	13.03 €
Bomba d'aigua	11.25 €
Base de l'hivernacle	20.00 €
Columnnes de l'hivernacle	6.00 €
Coberta de l'hivernacle	4.00 €
<i>LED</i>	1.9 €
Sensor de temperatura	1.6 €
Sensor de temperatura i humitat	5 €
Resistència	1.2 €
Plantes i terra	4.5 €
Cable	6 €
Interruptor	1 €
<b>Preu total</b>	<b>95.68 €</b>

Figura 1.3 Materials emprats en la fabricació del Treball de Recerca.

### 1.3-COSTOS DERIVATS DE L'ELABORACIÓ DE LA MEMÒRIA

El total de nombre d'hores emprades per realitzar la memòria del treball és de 62 hores. Com que el projecte ha requerit un total de 62 hores, el preu total s'obtindrà multiplicant el nombre d'hores emprades pel preu/hora. Aquest preu és de 3.224 €.

### 1.4-COST TOTAL ENGINYERIA

El cost total del Treball de Recerca és la suma dels costos derivats del disseny i dels costos de l'elaboració de la memòria del treball. Com podem observar a la figura 1.4, el cost total de l'enginyeria és de 4.992 €.



Costos derivats del	Hores	€/h	Cost total
Disseny	34	52 €	1.768 €
Elaboració de la memòria	62	52 €	3.224 €
<b>Total</b>			<b>4.992 €</b>

Figura 1.4 Càlcul del cost total de l'enginyeria.

## 2.-COSTOS DERIVATS DEL MUNTATGE

### 2.1-CÀLCUL DEL PREU HORA

Per fer el càlcul del preu hora suposarem que el projecte ha estat realitzat per un operari qualificat, treballant d'autònom.

Com passava amb els costos de l'enginyeria, aquests costos es divideixen en costos directes i indirectes.

#### 2.1.1-COSTOS DIRECTES

Els costos directes són aquells relacionats amb el salari de l'operari. Suposarem que el nostre treballador autònom cobra 18 €/h.



## 2.1.2-COSTOS INDIRECTES

Els costos indirectes són les despeses generals: lloguer del local, màquines i eines, materials del taller i les despeses generals.

Lloguer del local	2.600 €
Màquines i eines	900 €
Material del taller	700 €
Despeses generals	600 € 1000
<b>Costos indirectes totals</b>	<b>4.800 €</b>

Figura 2.1 Variables a tenir en compte per al càlcul dels costos indirectes.

Els costos derivats del muntatge ascendeixen a uns 4.800 €. Per a calcular el preu/hora que suposen els mateixos es consideren 1.800 h de treball i aplicant el coeficient del 65%, el cost aproximat és de 4 €/h.

## 2.1.3-COSTOS TOTALS DERIVATS DEL MUNTATGE

Sumant els costos directes i els indirectes el preu total del muntatge és de 22 €.

## 2.2-COST TOTAL DEL MUNTATGE DE L'HIVERNACLE

A la següent taula es detalla el nombre d'hores emprades en la construcció de la maqueta de l'hivernacle.



Tasques realitzades	Duració (hores)
Compra dels materials	5
Tallar les columnes	0.45
Muntar la bomba d'aigua	4
Muntatge de l'hivernacle	34
Automatització de l'hivernacle	10
<b>Total</b>	<b>53.45</b>

Figura 2.2 Taula resum de les hores emprades en les tasques realitzades.

A partir d'aquestes dades es conclou que per realitzar el muntatge de la maqueta són necessàries 53.45 h. El cost del muntatge de la maqueta ascendeix a 1175.9 €.

## ANNEX 2: PROGRAMA

# L'HIVERNACLE INTEL·LIGENT



/\*                                   Comunicación bluetooth ARDUINO-ANDROID (App Inventor)

Programado por: Laura Ysla

15 de gener 2015

\*/

// Este programa permite conectar la placa Arduino con una aplicación Android mediante bluetooth.

// Permite controlar el encendido de un LED (pin 13) accionando los botones de encendido y apagado de una aplicación Android.

// Permite visualizar en dicha aplicación Android los valores analógicos de temperatura y humedad sensados por el DHT11.

/\* ----- Declaración de variables -----\*/

// Librerías necesarias

**#include <SoftwareSerial.h>**     // Librería que permite la comunicación serie con diferentes pines de Arduino

**#include "DHT.h"**               // Librería sensor DHT11

//Asignamos los pines de conexión.

**#define txPin 3**     // Pin de comunicación TX (txPin) del bluetooth

**#define rxPin 2**     // Pin de comunicación RX (rxPin)del bluetooth

**#define DHTPIN 4**    // Pin de lectura de datos del sensor de Temperatura y humedad DHT11



## L'HIVERNACLE INTEL·LIGENT



```
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
```

// En este apartado debemos definir el resto de pines conectados a la placa. Es importante asegurarse de que no tenemos ningun puerto duplicado.

```
SoftwareSerial bluetooth(rxPin, txPin); // el puerto nuevo de comunicacion se llamar bluetooth e indico los pines de comunicaci3n
```

```
char NOMBRE[10] = "TEST1"; // asignamos un nombre a nuestro modulo bluetooth para cuando lo veamos en linea
```

```
char BPS = '4'; // 1=1200 , 2=2400, 3=4800, 4=9600, 5=19200, 6=38400, 7=57600, 8=115200
```

```
char PASS[10] = "1234"; // password para realizar la conexi3n. Sino funciona 0000
```

```
char INBYTE;
```

```
int ledPin = 13; // led verd
```

```
int pulsador = 7; //led blanc pulsador
```

```
int ledblanc = 12; //led blan pulsador
```

```
int estado = LOW; //led blanc pulsador
```

```
int LDR = 0; //LDR_led blanc
```

```
int LDRValue = 0; //LDR_led blanc
```

```
int light_sensitivity = 750; //LDR_led blanc -- canviar aqui el valor!!
```

```
int LED = 11; //LDR_led blanc
```

```
const int LM35 = 1; //sensor de temperatura- led de colores
```

```
const int ledV = 5; //sensor de temperatura- led de colores
```

```
const int ledR = 6; //sensor de temperatura- led de colores
```

```
long miliVolts; //sensor de temperatura- led de colores
```

```
long temperatura; //sensor de temperatura- led de colores
```

```
int brillo; //sensor de temperatura- led de colores
```





## L'HIVERNACLE INTEL·LIGENT



```
int ledverd = 10; //sensor de temperatura i humitat
int ledvermell = 9; //sensor de temperatura i humitat
int ledgroc = 8; //sensor de temperatura i humitat
int humedad = 2; //sensor de humedad
int bomba= 1; //sensor de humedad
int humedad1; //sensor de humedad

/* ----- Definimos las condiciones iniciales de trabajo -----
---*/

void setup() {

  Serial.begin(9600); // Instrucción que abre el puerto seria a la velocidad de
transmisión indicada en baudios

  dht.begin();      // inicialización del sensor DHT11

  Serial.println("Conectado"); //mensaje de comprobacion bluetooth (monitor serie)
  // pinMode(h, INPUT);

  bluetooth.flush();      // Borramos el buffer del serial para evitar errores
  pinMode(rxPin, INPUT); // Configuramos los pines del bluetooth
  pinMode(txPin, OUTPUT);

  bluetooth.begin(9600); // marcamos la velocidad del puerto bluetooth
  pinMode(ledPin, OUTPUT); //led verd
  pinMode(pulsador, INPUT); // led blanc pulsador
  pinMode(ledblanc, OUTPUT); //led blanc pulsador
  pinMode(LED, OUTPUT); //LDR_led blanc
  pinMode(ledV, OUTPUT); //sensor de temperatura- led de colores
  pinMode(ledR, OUTPUT); //sensor de temperatura- led de colores
  pinMode(ledverd, OUTPUT); //sensor de temperatura i humitat
  pinMode(ledvermell, OUTPUT); //sensor de temperatura i humitat
```



```
pinMode(ledgroc, OUTPUT); //sensor de temperatura i humitat  
}
```

```
/* ----- Programa principal -----*/
```

```
// El programa principal consiste en llamar a los programas individuales que  
hemos creado.
```

```
// Es un bucle infinito que va ejecutando uno tras otro dichos programas.
```

```
void loop()
```

```
{
```

```
  sensor ();    // Llama al programa de lectura de sensores
```

```
  delay (5);    // espera de 5 ms
```

```
  bluetooth_app (); // Llama al programa de control de LEDs actuando sobre la  
aplicación Android
```

```
  delay (5);
```

```
  led_verd(); //Llama al programa de encender el led verde (fijo)
```

```
  delay (5);
```

```
  pulsador(); //Llama al programa de controlar el led blanco interior con el  
pulsador
```

```
  delay (5);
```

```
  sensor_llum(); //Llama al programa del sensor de luz (led blanc exterior)
```

```
  delay (5);
```

```
  sensor_LM35(); //Llama al programa del sensor de temperatura (led multicolor)
```

```
  delay (5);
```

```
  sensor_temperatura_humitat(); //Llama al programa de sensor DHT11 (leds:  
verd, vermell, groc)
```

```
  delay (5);
```

```
  sensor_humitat(); //Llama al programa del sensor de humedad casolà (bomba de  
agua)
```



```
delay (5);
```

```
}
```

```
/* ----- Programa control LED Android -----*/
```

```
// Este programa permite controlar el encendido y apagado de LEDs mediante la aplicación Android
```

```
// Se trata de repetir el proceso para el LED del pin 13 con el resto de LEDs que queramos controlar.
```

```
// En la aplicación Android crearemos los botones necesarios para controlar todos los LEDs.
```

```
// Vinculamos cada botón a un número entero (INBYTE) que enviaremos desde la aplicación.
```

```
void bluetooth_app()
```

```
{
```

```
    //led blanc interior
```

```
    INBYTE = bluetooth.read();    // lee la comunicacion bluetooth
```

```
    if( INBYTE == '2' )
```

```
    {
```

```
        digitalWrite(ledblanc, LOW); // si pulsas el numero 3 escribe en el led apagado
```

```
    }
```

```
    if( INBYTE == '1' )
```

```
    {
```

```
        digitalWrite(ledblanc, HIGH); // si pulsas el numero 4 escribe led encendido
```

```
    }
```

```
    // bomba d'aigua
```

```
    if (INBYTE == '4' )
```

## L'HIVERNACLE INTEL·LIGENT



```
{  
    digitalWrite(bomba, LOW); //si pulsas el numero 4 escribe en la bomba  
    apagada  
}  
  
if ( INBYTE == '3' )  
{  
    digitalWrite(bomba, HIGH); //si pulsas 3 escribe en la bomba encendida  
}  
}
```

// Aquí repetiremos el código utilizado para el led 13.

// Debemos vincular cada LED con dos botones de la aplicación y, por tanto, con dos números enteros que darán la orden de encendido y apagado.

```
/* ----- Programa sensores -----*/
```

// Este programa se encarga de leer los valores analógicos de los sensores utilizados y de enviar los datos via Bluetooth.

// La salida de datos para su posterior visualización es única, por tanto es necesario que se realice en el orden preciso.

// Añadir aquí la lectura de temperatura exterior y posterior visualización

```
void sensor()  
{  
    int h_int= dht.readHumidity(); // Lectura de la humedad interior (variable tipo  
    entero h_int)  
    int T_int = dht.readTemperature(); // Lectura de la temperatura  
    interior(variable tipo entero T_int)  
    int temperatura;  
    int humedad1;
```



## L'HIVERNACLE INTEL·LIGENT



```
// Programa de lectura de datos del sensor LM35

    bluetooth.print ("Temperatura interior: ");           // Texto a visualizar (sin cambio
de línea)

    bluetooth.print (T_int);                               // Valor de temperatura adquirido (sin
cambio de línea)

    bluetooth.println (" °C");                             // Unidades a visualizar (+ cambio de
línea)

    bluetooth.print ("Humedad interior: ");

    bluetooth.print (h_int);

    bluetooth.println (" %");

    // repetimos la estructura para la visualización de la temperatura exterior

    bluetooth.print ("Temperatura exterior: ");           // Texto a visualizar (sin cambio
de línea)

    bluetooth.print (temperatura);                         // Valor de temperatura adquirido (sin
cambio de línea)

    bluetooth.println (" °C");                             // Unidades a visualizar (+ cambio de
línea)

// sensor de humitat casolà

    bluetooth.print ("Humedad de la tierra: ");

    bluetooth.print (humedad1);

    bluetooth.println (" %");

    delay (500);                                           // el programa espera 0,5 segundos para realizar la
próxima medida de valores de humedad y temperatura

}

// Resta de programes

// Programa led verd que està encès sempre
```



## L'HIVERNACLE INTEL·LIGENT



```
void led_verd()
{
    digitalWrite(ledPin, HIGH); // led verd
}

// Programa pulsador. Controla el led blanc interior
void pulsador()
{
    estado = digitalRead(pulsador); //led blanc pulsador
    digitalWrite(ledblanc, estado); //led blanc pulsador

for ( int i = 0; i < 100; i++) //led blanc pulsador
{
    delay(10); //led blanc pulsador
}
}

//Programa sensor LDR. Encén el led blanc exterior
void sensor_llum()
{
    LDRValue = analogRead(LDR); //LDR_led blanc
    Serial.println(LDRValue); //LDR_led blanc
    delay(50); //LDR_led blanc
    if (LDRValue < light_sensitivity) //LDR_led blanc
    {
        digitalWrite(LED, HIGH); //LDR_led blanc
    }
    else
```



## L'HIVERNACLE INTEL·LIGENT



```
{
  digitalWrite(LED, LOW); //LDR_led blanc
}
}

// Programa del sensor de temperatura. Encén el led multicolor de la part exterior
void sensor_LM35()
{
  miliVolts = (analogRead(LM35) * 5000L) / 1023; //sensor de temperatura- led de
  colores
  temperatura = miliVolts / 10; //sensor de temperatura- led de colores
  Serial.print("Temperatura: "); //sensor de temperatura- led de colores
  Serial.print(temperatura); //sensor de temperatura- led de colores
  Serial.println(" graus"); //sensor de temperatura- led de colores
  delay(200); //sensor de temperatura- led de colores
  brillo = map(temperatura, 5, 40, 0, 255); //sensor de temperatura- led de colores
  brillo = constrain(brillo, 0, 255); //sensor de temperatura- led de colores
  analogWrite(ledR, brillo); //sensor de temperatura- led de colores
  analogWrite(ledV, 255 - brillo); //sensor de temperatura- led de colores
}

//Programa del sensor de temperatura i humitat (DHT11) (leds: verd, vermell,
groc)
void sensor_temperatura_humitat()
{
  float h = dht.readHumidity(); //sensor de temperatura i humitat
  float t = dht.readTemperature(); //sensor de temperatura i humitat
  Serial.print("Humedad: "); //sensor de temperatura i humitat
```





```
Serial.print(h); //sensor de temperatura i humitat
Serial.print(" % "); //sensor de temperatura i humitat
Serial.print("Temperatura: "); //sensor de temperatura i humitat
Serial.print(t); //sensor de temperatura i humitat
Serial.println (" °C"); //sensor de temperatura i humitat
delay(2000);

if (t > 0){ //sensor de temperatura i humitat
    digitalWrite(ledverd, HIGH); //sensor de temperatura i humitat
    digitalWrite(ledvermell, LOW); //sensor de temperatura i humitat
}

if (t > 20){ //sensor de temperatura i humitat
    digitalWrite(ledvermell, HIGH); //sensor de temperatura i humitat
    digitalWrite(ledverd, LOW); //sensor de temperatura i humitat
}

if (h < 50){ //sensor de temperatura i humitat
    digitalWrite(ledgroc, HIGH); //sensor de temperatura i humitat
    delay(500); //sensor de temperatura i humitat
    digitalWrite(ledgroc,LOW); //sensor de temperatura i humitat
    delay(50); //sensor de temperatura i humitat
}

}

// Programa de la bomba d'aigua. Sensor de humitat casolà
void sensor_humitat()
{
    Medir_Humedad(); //sensor de humedad
    char dato = Serial.read(); //sensor de humedad
```



## L'HIVERNACLE INTEL·LIGENT



```
if (dato=='1') //sensor de humedad
{
    Mostrar_humedad(); //sensor de humedad
}
else {} //sensor de humedad
}

void Medir_Humedad() //sensor de humedad

{
    humedad1 = analogRead(0); //sensor de humedad
if(humedad1 >= 1000) { //sensor de humedad
    digitalWrite(bomba, HIGH); //sensor de humedad
}
else { //sensor de humedad
    digitalWrite(bomba, LOW); //sensor de humedad
}
}

void Mostrar_humedad() //sensor de humedad
{
    Serial.print("Humedad:"); //sensor de humedad
    Serial.println(analogRead(0)); //sensor de humedad
}
```



L'HIVERNACLE INTEL·LIGENT L'HIVERNACLE INTEL·LIGENT L'HIVERNACLE INTEL·LIGENT L'HIVERNACLE

